Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1989

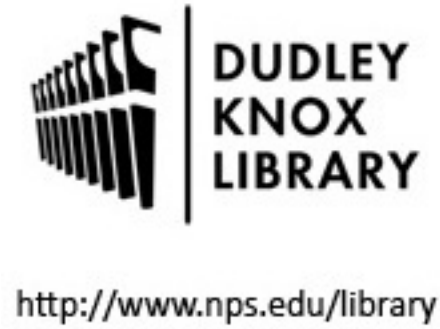
A simulation of optical propagation through atmospheric turbulence using two-dimensional Fourier transform techniques

Turner, Jeffrey L.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/26244

Downloaded from NPS Archive: Calhoun



http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5002

NPS 61-089-012

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

T9546

A SIMULATION OF OPTICAL PROPAGATION THROUGH ATMOSPHERIC TURBULENCE USING TWO-DIMENSIONAL FOURIER TRANSFORM TECHNIQUES

by

Jeffrey L. Turner

June 1989

Thesis Advisor: Donald L. Walters

Approved for public release; Distribution is unlimited.

Prepared for:

Weapons Laboratory/ARCA
Kirtland Air Force Base,
New Mexico 87117

T246013

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. C. Austin
Superintendent

Harrison Shull
Provost

This report was prepared in conjunction with research conducted for the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico.

Release by:

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NPS 61-089-012	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 61	7a NAME OF MONITORING ORGANIZATION Weapons Laboratory
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) ARCA Kirtland Air Force Base New Mexico 87117
8a NAME OF FUNDING / SPONSORING ORGANIZATION Weapons Laboratory 8b OFFICE SYMBOL (If applicable) ARCA	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFWL 87-MP-240
8c ADDRESS (City, State, and ZIP Code) Kirtland Air Force Base New Mexico 87117	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) A Simulation Of Optical Propagation Through Atmospheric Turbulence Using Two-Dimensional Fourier Transform Techniques

12 PERSONAL AUTHOR(S)
Turner, Jeffrey L. in conjunction with Donald L. Walters

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	June 1989	79

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Optical propagation, Atmospheric turbulence, Coherence length, Huygens-Fresnel theory

19 ABSTRACT (Continue on reverse if necessary and identify by block number) Understanding turbulence degradation of electromagnetic wave propagation is essential for efficient operation of laser weapons, target designators, and imaging systems. Random atmospheric refractive index inhomogeneities alter the phase and amplitude of electromagnetic waves.

This thesis attempts to model atmospheric turbulence effects by using filtered Gaussian phase screens to represent the random nature of refractive index changes. The simulation uses two-dimensional 512 x 512 fast Fourier transform (FFT) techniques with extended Huygens-Fresnel principles performed on a desk top computer.

Simulation verification was accomplished by comparing calculated and theoretical spatial coherence lengths, ρ_o. Phase only screens produced coherence lengths that were 30% larger than theoretical values. By using random phase and amplitude screens, the calculated coherence lengths

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Donald L. Walters	22b TELEPHONE (Include Area Code) (408) 646-2267 22c OFFICE SYMBOL 61We

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

Block 19 Continued

agreed to within 3 % of theoretical values. Saturation of the normalized intensity variance, σ^2/I^2 , occurred for increasing turbulence using a single phase-amplitude screen.

Approved for public release; distribution is unlimited.

A Simulation Of Optical Propagation Through Atmospheric Turbulence Using Two-Dimensional Fourier Transform Techniques

by

Jeffrey L. Turner
Lieutenant, United States Navy
B.S., Stephen F. Austin State University, 1983

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
June 1989

ABSTRACT

Understanding turbulence degradation of electromagnetic wave propagation is essential for efficient operation of laser weapons, target designators, and imaging systems. Random atmospheric refractive index inhomogeneities alter the phase and amplitude of electromagnetic waves.

This thesis attempts to model atmospheric turbulence effects by using filtered Gaussian phase screens to represent the random nature of refractive index changes. The simulation uses two-dimensional 512 x 512 fast Fourier transform (FFT) techniques with extended Huygens-Fresnel principles performed on a desk top computer.

Simulation verification was accomplished by comparing calculated and theoretical spatial coherence lengths, ρ_0. Phase only screens produced coherence lengths that were 30% larger than theoretical values. By using random phase and amplitude screens, the calculated coherence lengths agreed to within 3% of theoretical values. Saturation of the normalized intensity variance, σ^2/I^2, occurred for increasing turbulence using a single phase-amplitude screen.

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKGROUND 3

A. STATISTICAL DESCRIPTION OF ATMOSPHERIC TURBULENCE 3

1. Stationarity, Homogeneity, and Isotropy 3

2. The Structure Function 4

3. The Correlation Function and Power Spectral Density 6

B. ELECTROMAGNETIC PROPAGATION THROUGH TURBULENCE ... 7

1. The Wave Equation 7

2. The Born Approximation 8

3. The Rytov Approximation 10

C. HUYGENS-FRESNEL THEORY 12

D. MUTUAL COHERENCE FUNCTION 13

E. DIFFRACTION INTEGRAL 14

III. NUMERICAL SIMULATION MODEL 17

A. HARDWARE AND SOFTWARE SPECIFICS 17

B. REQUIRED ALGORITHMS 18

1. Random number generator 18

2. Fast Fourier Transform 20

C. SIMULATION DESCRIPTION 23

1. User defined quantities 23

2. Aperture Mutual Coherence Function 24

3. Phase screen generation.....24
a. Classical method.....25
b. Martin and Flatte method.....25
4. Filtering.....26
5. Propagation.....29
6. Atmospheric Mutual Coherence Function.....30
7. Coherence Length.....31
IV. RESULTS.....32
A. APERTURE MCF.....32
B. FILTERING.....35
C. ATMOSPHERIC MCF.....35
D. COHERENCE LENGTH.....40
E. INTENSITY VARIANCE SATURATION.....51
V. CONCLUSIONS.....53
APPENDIX A. RANDOM NUMBER GENERATOR SUBROUTINE.....55
APPENDIX B. GAUSSIAN DISTRIBUTION SUBROUTINE.....56
APPENDIX C. SIMULATION CODE.....57
LIST OF REFERENCES.....69
INITIAL DISTRIBUTION LIST.....70

ACKNOWLEDGEMENT

I would like to dedicate this thesis to my loving wife Terri, who gave me the strength and support to complete this work. I would also like to thank my mother, Dorothy Dunnavant for encouraging me from the start.

I. INTRODUCTION

Turbulence in the atmosphere has a detrimental effect on the propagation of electromagnetic waves at optical wavelengths. Minute changes in the refractive index of the atmosphere along the path of the wave cause perturbations in the wavefront in both intensity and phase. Consequently, random atmospheric density fluctuations degrade the performance of optical imaging systems. Understanding this phenomenon is essential for future endeavors in laser weapons and designators, as well as imaging systems.

This thesis attempts to model the effects of atmospheric turbulence using the extended Huygens-Fresnel principle. This theory considered both Fraunhofer and Fresnel propagation although the simulation focused on a single phase screen using the Fraunhofer approximation.

A Rytov approximation filtered, Gaussian distributed, random phase and amplitude screen simulated the effects of turbulence on an optical beam. The magnitude of turbulence of this screen was proportional to the index of refraction structure parameter, C_n^2.

The simulation code used two-dimensional fast Fourier Transform (FFT) algorithms extensively to model optical wave propagation in the Fraunhofer regime.

Accuracy of the simulation was checked by comparison of calculated and theoretical coherence lengths, ρ_0, which were obtained from the e^{-1} point of the atmospheric mutual coherence function (MCF). Another check verified that intensity variance saturation occured for an increase in turbulence, as measured experimentally and predicted by theory.

II. BACKGROUND

A. STATISTICAL DESCRIPTION OF ATMOSPHERIC TURBULENCE

1. Stationarity, Homogeneity, and Isotropy

Most theorems involving random processes require that the functions satisfy certain restrictions. "Stationarity" implies that the mean value of the random function does not change with time. "Homogeneity" implies that translations in x, y, and z coordinates (a Galilean coordinate transformation) do not affect variables of the random function. "Isotropy" means that the random function is independent of any rotational coordinate transformation. [Ref. 1]

The last two conditions imply that the statistical representation of the random function for two points $f(\bar{r}_1)$ and $f(\bar{r}_2)$ depend only on the magnitude of their separation $| \bar{r}_1 - \bar{r}_2 |$. [Ref. 1]

If these three restrictions only hold for a local volume of space, L^3, the random field is said to be locally stationary, locally homogeneous, and locally isotropic for $| \bar{r}_1 - \bar{r}_2 | < L$.

Turbulence of the atmosphere is a process that can be best described by a random function. Minimal violations of stationarity, homogeneity, and isotropy occur if the neighborhood is sufficiently small, less than some outer scale length L. [Ref. 1]

2. The Structure Function

Because of their nature, random functions, f, can best be described stochastically. The simplest moment of f is the mean value, $\langle f \rangle$.

Another common parameter used to represent a random function is the variance, σ^2, defined as

$$\sigma^2 = \langle \, (\, f(\bar{r}) - \langle f(\bar{r}) \rangle \,)^2 \, \rangle . \qquad (2.1)$$

In a physical random field, such as atmospheric turbulence, the mean value, $\langle f(\bar{r}) \rangle$, is difficult to ascertain because of trends in the mean as well as variations in spatial position, vertically and horizontally. Therefore, the mean and variance are awkward parameters for describing a real, random, atmospheric field. To resolve this difficulty, Kolmogorov introduced the structure function

$$D_f(r) = D_f(\bar{r}_2 - \bar{r}_1) = \langle [f(\bar{r}_1) - f(\bar{r}_2)]^2 \rangle , \qquad (2.2)$$

which resembles the variance at first glance. [Ref. 2] The structure function is an ensemble average taken over all possible point pairs $\bar{r}_1$ and $\bar{r}_2$. [Ref. 2]

Through dimensional analysis, and assuming isotropy, Kolmogorov deduced a relation between the structure function and the distance between the sampling points,

$$D_f(r) = C_f^2 \, r^{2/3} \qquad (2.3)$$

which holds over a limited volume, called the inertial subrange. [Ref. 2] Although developed for the velocity field $\bar{v}$, Equation 2.3 also holds for certain atmospheric functions called conservative passive additives.

Conservative passive additives are quantities that play no part in the turbulence of the medium. Temperature is not such a quantity, but potential temperature, which corrects for the adiabatic change in temperature with altitude, is. Therefore, the relation

$$D_T(r) = C_T^2 \, r^{2/3} \tag{2.4}$$

holds within the inertial subrange where T is the potential temperature.

In examining problems of optical propagation, the index of refraction is the essential parameter. To express the index of refraction as a conservative passive additive, it is useful to manipulate the following relation

$$n = 1 + 77.6(1 + 7.52 \times 10^{-3}/\lambda^2) \, (P/T) \times 10^{-6} \tag{2.5}$$

where λ is the wavelength of the radiation in μm, P is the atmospheric pressure in mb, and T is the atmospheric temperature in K. [Ref. 3] The differential of Equation 2.5 is

$$dn = - 79 \times 10^{-6} \, P \, dT/T^2 \tag{2.6}$$

for red light (λ = 0.6 μm) after ignoring the dP term (isobaric turbulence). Since potential temperature is a passive additive, the index of refraction is also. Using Equation 2.6 to express the index of refraction structure parameter in terms of the temperature structure parameter gives

$$D_n(r) = C_n^2\, r^{2/3} = (79\ P/T^2 \times 10^{-6})^2\, C_T^2\, r^{2/3} \qquad (2.7)$$

[Ref. 3].

3. The Correlation Function and Power Spectral Density

Expanding Equation 2.2 gives

$$D_f(r) = \langle f^2(\bar{r}_1)\rangle - 2\langle f(\bar{r}_1) f(\bar{r}_2)\rangle + \langle f^2(\bar{r}_2)\rangle. \qquad (2.8)$$

But if we assume an homogeneous medium, then $\langle f^2(r) \rangle$ is the same for any point r, assuming $\langle f(r)\rangle = 0$. The structure function becomes

$$D_f(r) = 2\, B_f(0) - 2\, B_f(r), \qquad (2.9)$$

where Tatarski defined the correlation function, B_f, as

$$B_f(r) = \langle\, f(\bar{r}_1)\ f^*(\bar{r}_2)\,\rangle \qquad (2.10)$$

in a stationary, homogeneous, and isotropic medium. [Ref. 2]

Using stochastic Fourier-Stieltjes integrals, Tatarski developed a relation for the three-dimensional power spectral density function, $\Phi(f)$, as the Fourier transform of the correlation function. Writing this in freqency (f) form as opposed to radian freqency (K) form preferred by Tatarski,

$$\Phi(f) = \int_V \exp(-2\pi i \bar{f}\cdot\bar{r})\, B(r)\, d^3\bar{r}, \qquad (2.11)$$

and conversely,

$$B_f(r) = \int_V \exp(-2\pi i \bar{f}\cdot\bar{r})\, \Phi(f)\, d^3\bar{f}. \qquad (2.12)$$

[Ref. 2]

Using the fact that the correlation function is even, and using relations (2.7) and (2.9), the structure function for the Kolmogorov power spectral density is

$$\Phi_n(f) = 1.303\, C_n^2\, f^{-11/3}, \qquad (2.13)$$

for the Kolmogorov power spectral density. This equation is analogous to

$$\Phi_n(K) = 0.033\, C_n^2\, K^{-11/3}, \qquad (2.14)$$

an expression developed by Tatarski in K space that is seen in many publications. [Ref. 2]

B. ELECTROMAGNETIC PROPAGATION THROUGH TURBULENCE

1. The Wave Equation

Central to any study of electromagnetic propagation are the four equations of Maxwell, presented here in Gaussian units

$$\nabla \cdot \bar{H} = 0, \qquad (2.15)$$

$$\nabla \times \bar{H} = -ikn^2\bar{E}, \qquad (2.16)$$

$$\nabla \cdot (n^2 \bar{E}) = 0, \tag{2.17}$$

$$\nabla \times \bar{E} = ik\bar{H}, \tag{2.18}$$

for the assumptions that the medium has zero conductivity, unit magnetic permeability, and sinusoidal dependence.

Taking the curl of Equation 2.18 gives

$$\nabla \times (\nabla \times \bar{E}) = \nabla \times (ik\bar{H}). \tag{2.19}$$

Using a vector identity and Equation 2.16, Equation 2.19 becomes

$$-\nabla^2 \bar{E} + \nabla(\nabla \cdot \bar{E}) = k^2 n^2 \bar{E}. \tag{2.20}$$

Expanding Equation 2.17 gives a relation for $\nabla \cdot \bar{E}$, which when inserted into Equation 2.20 gives the wave equation

$$\nabla^2 \bar{E} + k^2 n^2 \bar{E} + 2\nabla[\bar{E} \cdot \nabla(\log n)] = 0. \tag{2.21}$$

If the wavelength of the electromagnetic wave is small compared to the dimensions of the refractive inhomogeneities, then the $2\nabla[\bar{E} \cdot \nabla(\log n)]$ term is negligible. In this case, the wave equation reduces to the simple form

$$\nabla^2 \bar{E} + k^2 n^2 \bar{E} = 0. \tag{2.22}$$

[Ref. 3]

2. The Born Approximation

One method of solving the wave equation, employed by both Tatarski [Ref. 2] and Clifford [Ref. 3], is the method of small perturbations or the Born approximation. This method

expands the electric field and index of refraction as a power series

$$E = E_0 + E_1 + \cdots, \tag{2.23}$$

$$n = 1 + n_1 + \cdots, \tag{2.24}$$

where E_0 is a constant in time, and E_1 and n_1 are time varying.

Inserting these two relations into Equation 2.22 and equating same order terms gives

$$\nabla^2 E_0 + k^2 E_0 = 0, \tag{2.25}$$

$$\nabla^2 E_1 + k^2 E_1 + 2n_1 k^2 E_0 = 0, \tag{2.26}$$

ignoring all second and higher order terms.

Assuming that the electromagnetic wave propagates in the z-direction and $E_0 = \exp(ikz)$, then Equation 2.26 becomes

$$\nabla^2 E_1 + k^2 E_1 + 2n_1 k^2 \exp(ikz) = 0. \tag{2.27}$$

Clifford solved this wave equation for E_1 by utilizing a Green's function. The solution is the convolution of a plane wave Green's function with the source term, shown here in scalar form

$$E_1(\bar{r}) = \frac{1}{4\pi} \int_V \frac{\exp[ik|\bar{r}-\bar{r}'|]}{|\bar{r}-\bar{r}'|} [2k^2 n_1(\bar{r}') \exp(ikz')] \, d^3\bar{r}, \tag{2.28}$$

where $\bar{r}$ is the position of a specific point in the image plane and $\bar{r}'$ is the postion of specific source point. [Ref. 3]

Figure (2.1) shows the coordinate system used to express the solution to the wave equation for propagation between planes $\bar{z}'$ and $\bar{\rho}'$ to z and ρ.

For normal propagation situations, $|\bar{z}-\bar{z}'| = R >> |\bar{\rho}-\bar{\rho}'|$ so that Equation 2.28 expands into the form

$$E_1(\bar{r}) = \frac{1}{4\pi}\int_v \frac{\exp\{ikR\ [1 + |\bar{\rho}-\bar{\rho}'|^2/2R^2 + \cdots]\}}{R\ [1 + |\bar{\rho}-\bar{\rho}'|^2/2R^2 + \cdots]} \times [2k^2\ n_1(\bar{r}')\ \exp(ikz')]\ d^3\bar{r}', \qquad (2.29)$$

using the binomial expansion. Dropping those terms consistent with the Fresnel approximation in Huygens-Fresnel optics theory, this equation reduces to

$$E_1(\bar{r}) = \frac{k^2\ \exp(ikz)}{2\pi R}\int_v \exp\{ik|\bar{\rho}-\bar{\rho}'|^2/2R\}\ n_1(\bar{r}')\ d^3\bar{r}', \qquad (2.30)$$

which is known as the Fresnel diffraction formula. [Ref. 3]

3. The Rytov Approximation

Another method of solving the wave equation, also used by both Tatarski and Clifford, is the method of smooth perturbations or the Rytov approximation. This technique assumes that the electric field is of the form

$$E = \exp(\psi) = \exp(X+iS) = A\ \exp(iS). \qquad (2.31)$$

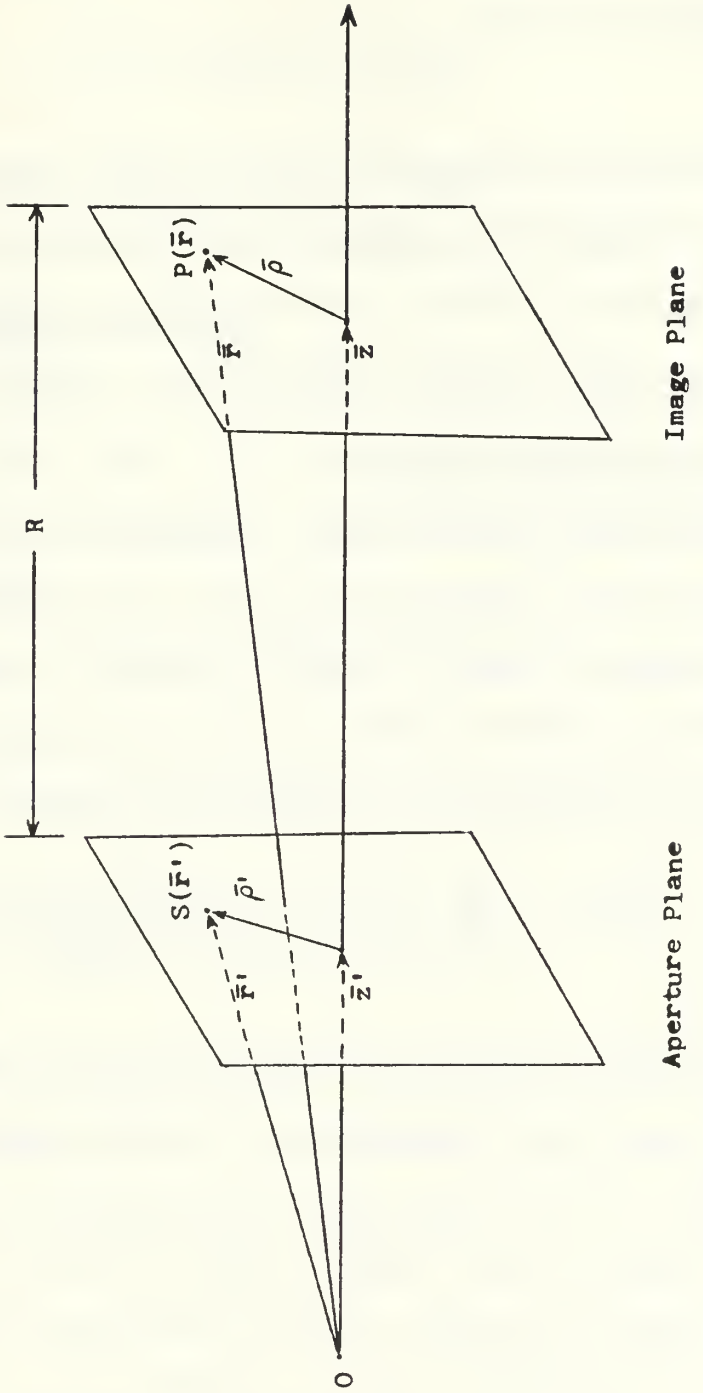


Figure 2.1 Aperture and Image Plane Geometry.

Using power series expansions for the electric field, Equations 2.25 and 2.26 still hold. Placing Equation 2.31 into these two gives

$$(\nabla^2 E)/E + k^2 n^2(\bar{r}) = 0. \qquad (2.32)$$

Tatarski shows that this method gives the same results as the method of small perturbations but the range of validity is larger than for the Born approximation. Equation 2.26 holds as long as $\nabla \cdot \bar{E}$ is small compared to λ. [Ref. 2]

C. HUYGENS-FRESNEL THEORY

Another way of obtaining a relation for the electric field amplitude at the observation plane, is by extending the ideas of the Huygens-Fresnel theory to include a random phase term. The Huygens-Fresnel theory offers this solution for the electric field seen in the observation plane

$$E(\bar{r}) = -\frac{ik}{2\pi} \int_S E(\bar{r}') \frac{\exp(ik|\bar{r}-\bar{r}'|)}{|\bar{r}-\bar{r}'|} d\rho'. \qquad (2.33)$$

[Ref. 5]

Lutomirski and Yura develop an extension of the Huygens-Fresnel idea which incorporates a random phase term which is equivalent to the form used in the Rytov approximation, $\exp(\psi)$, where ψ is a complex phase. [Ref. 4] This extended integral is

$$E(\bar{r}) = -\frac{ik}{2\pi} \int_S E(\bar{r}') \exp(\psi(\bar{r})) \frac{\exp(ik|\bar{r}-\bar{r}'|)}{|\bar{r}-\bar{r}'|} d\rho'. \qquad (2.34)$$

In the geometrical optics limit, $\psi(r)$ becomes the optical phase

$$\psi(r) = ik \int n_1(\bar{z})\, dz. \tag{2.35}$$

Expanding ψ into a power series, Equation 2.35 reduces to Equation 2.30 as developed by Tatarski and Clifford.

D. MUTUAL COHERENCE FUNCTION

In trying to get a quantitative measure for turbulence, Lutomirski and Yura [Ref. 4] examined the average intensity of the wave at the observation point. The average intensity at $\bar{r}$ is defined as

$$\langle I(r) \rangle = \langle E(\bar{r}_1)\, E^*(\bar{r}_2) \rangle. \tag{2.36}$$

From Equation 2.34:

$$\langle I(r) \rangle = \Big\langle (k^2/4\pi^2) \iint E(\bar{r}_1')E^*(\bar{r}_2') \exp[\psi(\bar{r}_1) + \psi^*(\bar{r}_2)] \times \frac{\exp[ik(|\bar{r}_1-\bar{r}_1'| - |\bar{r}_2-\bar{r}_2'|)]}{|\bar{r}_1-\bar{r}_1'|\,|\bar{r}_2-\bar{r}_2'|}\, d\rho_1\, d\rho_2 \Big\rangle \tag{2.37}$$

Picking out only the time-dependent portion of this result, the entire integral reduces to evaluating

$$\langle \exp[\psi(\bar{r}_1) + \psi^*(\bar{r}_2)] \rangle. \tag{2.38}$$

This term is the atmospheric mutual coherence function (MCF) which contains the elements of the propagation through turbulence.

Lutomirski and Yura [Ref. 4] relate the atmospheric MCF to the wave structure function, D, by the relation

$$MCF(\rho) = \langle \exp[\psi(\bar{r}_1) + \psi^*(\bar{r}_2)] \rangle = \exp[-D(\rho)/2] \quad (2.39)$$

where $D(\rho) = D_X(\rho) + D_S(\rho)$ from the Rytov relation $\psi = X + iS$ or

$$MCF(\rho) = \exp[-(\rho/\rho_0)^{5/3}], \quad (2.40)$$

where

$$\rho_0 = [\, 1.46\, k^2 \int_0^R Cn^2\, dz\,]^{-3/5}. \quad (2.41)$$

and k is the wavenumber. Cn^2 is the refractive index structure parameter defined by Equation 2.7, and R is the distance of propagation through the atmosphere. The coherence length, ρ_0, is defined as the distance where transverse spatial coherence of the wave drops to the e^{-1}. [Ref. 1]

E. THE DIFFRACTION INTEGRAL

Like Lutomirski and Yura, Roberts [Ref. 6] begins with the Huygens-Fresnel integral for the propagation of light waves after making the Fresnel approximation ($|\bar{z}-\bar{z}'| = R >> |\bar{\rho}-\bar{\rho}'|$)

$$E(\bar{r}) = \frac{-ik}{2\pi R} \int_s d\rho'\, E(\bar{r}')\, \exp\{ik[R^2 + |\bar{\rho}' - \bar{\rho}|^2]^{1/2}. \quad (2.42)$$

In this expression, k is the wavenumber and the notation is the same as shown in Figure (2.1).

Employing the standard practice of expanding by the binomial expansion and neglecting smaller terms gives

$$E(\bar{r}) = \frac{-ik}{2\pi R} \exp[-ikR] \int_s d\rho'\, E(\bar{r}')\, \exp\{\frac{ik}{2R} |\bar{\rho}' - \bar{\rho}|^2\}. \quad (2.43)$$

Since the exponential term outside the integral doesn't affect intensity measurements, it can be neglected. Expanding the quadratic term gives the relation

$$E(\bar{r}) = -\frac{ik}{2\pi R}\int_s d\rho\,' \; E(\bar{r}') \exp\{\frac{ik}{2R}\,[\,\rho'^2 - 2\,\bar{\rho}\,'\cdot\bar{\rho} + \rho^2]\}$$

$$= -\frac{ik}{2\pi R}\exp[\frac{ik}{2R}\rho'^2]\int_s d\rho'\; E(r')\exp[\frac{ik}{2R}\rho^2]\exp[-\frac{ik\bar{\rho}}{R}\cdot\bar{\rho}\,'].$$

(2.44)

This is called the convolution form of the Fresnel diffraction integral which differs from the Fraunhofer approximation only by the quadratic phase term $\exp[\frac{ik\rho^2}{2R}]$. [Ref. 6]

If we denote the Fourier transform of $E(\bar{r})$ as $\tilde{E}(\bar{f})$, then

$$\tilde{E}(\bar{f}) = \int_s d\rho \;\; \exp(-2\pi i\bar{f}\cdot\bar{\rho}\,)\; E(\bar{r}). \qquad (2.45)$$

Inserting $E(\bar{r})$ as defined in Equation 2.43, gives

$$\tilde{E}(\bar{f}) = -\frac{ik}{2\pi R}\int_s d\rho\,' \; E(\bar{r}')\int_s d\rho \;\; \exp(-2\pi i\bar{f}\cdot\bar{\rho}\,)\exp\{\frac{ik}{2R}|\,\bar{\rho}\,'-\bar{\rho}\,|^2.$$

(2.46)

Changing variables to $\bar{q} = \bar{\rho}\,' - \bar{\rho}$, the electric field spectrum becomes

$$\tilde{E}(\bar{f}) = -\frac{ik}{2\pi R}\int_s d\rho\,' \; E(\bar{r}')\exp(-2\pi i\bar{f}\cdot\bar{\rho}\,)$$

$$\text{X}\quad\int_s dq\,\exp(-2\pi i\bar{f}\cdot\bar{q})\exp(\frac{ik}{2R}\,q^2). \qquad (2.47)$$

The q integral is shown to be the Fourier transform of the Gaussian function

$$\frac{2\pi iR}{k} \exp\left(-\frac{2\pi^2 iRf^2}{k}\right). \qquad (2.48)$$

[Ref. 6] Substituting this function into Equation 2.47 gives

$$\tilde{E}(\bar{f}) = \exp\left(-\frac{2\pi^2 iRf^2}{k}\right) \int d\rho\,' \; E(\bar{r}') \exp(-2\pi i \bar{f}\cdot\bar{\rho}\,), \qquad (2.49)$$

which is the inverse transform of what we sought,

$$E(\bar{r}) = \int df \exp(2\pi i \bar{f}\cdot\bar{r}) \exp\left(-\frac{2\pi^2 iRf^2}{k}\right)$$
$$\times \int d\rho\,' \; E(\bar{r}') \exp(-2\pi i \bar{f}\cdot\bar{\rho}\,) \qquad (2.50)$$

This is the transfer form of the Fresnel diffraction integral. [Ref. 6]

Equations 2.44 and 2.50 are two different forms of the diffraction integral that are useful for two types of wave propagation. The convolution form, Equation 2.44, is best suited for long distance propagation (Fraunhofer regime) since R is in the denominator of the exponential term. The transfer form, Equation 2.50, is best suited for short distance propagation since R is present in the numerator of the exponential term. [Ref. 6]

III. NUMERICAL SIMULATION

A. SIMULATION HARDWARE AND SOFTWARE SPECIFICS

This numerical simulation modeled the propagation of an electromagnetic wave through a random turbulent medium. The procedures for this simulation required the creation and manipulation of multiple two-dimensional, 512 by 512, single precision, phase screens, requiring a minimum memory of several megabytes.

A COMPAQ Deskpro 80386-20 computer configured with a 16 megabyte RAM and 64 megabyte hard drive met hardware requirements. The computer also had EGA-VGA graphics and both HP Laser Jet II and Panasonic KX-P1092i printers. A Weitek 1167 math coprocessor increased execution speeds of the 20 Mhz 80387 coprocessor by a factor of 3-4.

The software requirements for writing the simulation code were met with the MicroWay 32 bit NDP FORTRAN-386 compiler and the Phar Lap operating system extender. It was chosen over the Science Applications International Corporation's SVS Fortran-386 compiler which had 10-20% faster program execution times but numerous bugs present in the graphics routines.

Unfortunately, due to limitations of version 1.4e of the NDP Fortran-386 compiler, the full graphic capabilities of VGA were not supported, so EGA graphics were used for screen

displays. Hardcopy graphs and plots were accomplished using Grapher and Surfer programs from Golden Software.

B. REQUIRED ALGORITHMS

For the successful operation of such a model both the random number generator and the two-dimensional fast Fourier transform (FFT) routines needed speed and accuracy. These will be discussed in detail in the following sections.

1. Random number generator

The effects of turbulence on the propagation of an electromagnetic wave are of a random nature. To emulate this random nature required a pseudo-random number generator algorithm. The minimum criteria for a quality random number generator was that the products be: (1) sufficiently random, (2) reproducible, and (3) rapidly obtained.

The NDP FORTRAN compiler did not come equipped with a random number generator, so an algorithm of the linear congruent type especially designed for 32 bit machines was implemented as a subroutine attached to the main program. This had the benefit of not binding this portion of the code to a specific machine setup. This random number generator algorithm came from Dr. Milne, who obtained it from class notes prepared by Dr. Harrison [Ref. 7]. This algorithm appears as subroutine *RAN* (Appendix A) in the simulation code.

The first criteria, that the generator produce numbers sufficiently random, was hard to quantify. A series of tests

exercised the statistical independence of succeeding numbers. There are many such tests that exist, the most common of which is the "Chi-Square" test. The following is a list of all tests used to check the quality of the random number generator algorithm:

1. Frequency Test. This tests the uniformity of the sequence of numbers.

2. Serial Test. This tests the two-dimensional uniformity of the sequence of numbers.

3. Runs Up and Down Test. This tests to see how long sequences of random numbers are that either go successively up or down.

4. Lagged-Product Test. This tests for correlations between random numbers R_i and R_{i+j} where j is the given lag value. A lag of 3 is especially difficult to pass.

5. Repeat Test. This simple qualitative check is to determine how long the sequence of random numbers becomes before the first number is repeated.

6. Scatter Plot Test. This is another qualitative test where random numbers plotted on a x-y plane are visually checked for correlation.

These six tests were performed for a variety of seed values to determine which gave the best performance. Results of the first five tests for two good seeds can be found in Table 3.1.

Choosing the same initial seed value gave reproducibility of the random numbers. The algorithm's simple two lines of code assured rapid execution.

TABLE 3.1 RANDOM NUMBER GENERATOR TEST DATA

This table provides the results of statistical tests of the random number generator algorithm. The X^2 values came from the Chi-Square table in Bevington [Ref. 11]

The results of the Lagged Product test correspond to the theoretical mean, μ_T, the calculated mean, μ, the theoretical standard deviation, σ_T, and the calculated standard deviation.

Test	Seed	Degrees of Freedom	X^2	Probability
Frequency	45739	9	4.0	91 %
	94377	9	5.1	82 %
Serial	45739	20	16.6	70 %
	94377	20	10.2	96 %
Runs Up and Down	45739	6	11.7	2 %
	94377	5	2.4	80 %

Test	Seed	μ_T	μ	σ_T	σ
Lagged Product	45739	0.250	0.230	0.088	0.092
	94377	0.250	0.265	0.088	0.086

2. Fast Fourier Transform

Since the most often used algorithm in the simulation was the Fast Fourier Transform (FFT), efficiency was critical. Two different routines were compared.

The first FFT routine considered was a package offered by NDP of one- and two-dimensional machine language FFT routines designed for easy linking by the NDP FORTRAN compiler. This package had routines available for real and complex arrays.

The second FFT routine considered was a simple real array subroutine coded by Dr. Walters from a BASIC demonstration package provided by Infotek.

Each FFT routine was compared for numerical output and speed of operation for the one-dimensional case. As expected, all routines were virtually identical in numerical output. However, speed of operation differed widely, as seen in Table 3.2.

TABLE 3.2 FFT SPEED OF OPERATION COMPARISON

This table contains results of FFT execution time comparison between NDP machine language routines and Dr. Walters' subroutine. Since NDP FORTRAN does not have a timing function with sufficient accuracy, execution times listed are mean times found from several runs, timed by stopwatch.

Array Size	NDP Routines *CFFT*	*RFFT*	Walters' Routine w/o & w/ Weitek	
1-D field size	time(s)	time(s)	time(s)	time(s)
2^{13}	2.60	1.45	2.84	1.20
2^{14}	5.27	2.88	5.86	2.32
2^{15}	10.94	5.84	12.27	4.94
2^{16}	22.78	12.05	25.90	10.26
2^{17}	47.61	25.02	54.53	21.55

The differences in run time for the two NDP machine language routines were due to the *CFFT* routine being more cumbersome in converting repeatingly between real and complex arrays. The compiled subroutine was significantly faster than both *CFFT* and *RFFT* NDP machine language routines unless the Weitek was deactivated. Using a subroutine enclosed within

the main program had the additional benefit of not being machine dependent. Therefore, it was decided that Dr. Walters' subroutine would be used for all FFTs needed in the simulation code. Conversion of this subroutine for two-dimensional use was relatively easy. The NDP two-dimensional FFT routines never worked correctly.

A FFT "breaks down" a function into its cosine and sine constituents (real and imaginary terms, respectively). The range of spatial frequencies represented in a FFT depend on the array size chosen. The minimum frequency is $f_{min} = 1/W$ and the maximum frequency is $f_{max} = n/W$, where W is the physical width of the array and n is the number of sampling points across the array. When energy at frequencies near these cutoff values appeared, problems developed due to the discrete nature of the FFT.

One problem exists in representing spatial frequencies smaller than f_{min} due to the high amplitude, low frequency "tilt" of the $f^{-11/3}$ filtered wavefront. Since a tilted straight line approximates a cosine or sine function for only a small region, it helps to make the wavefront smaller than the FFT array size. Hence, to help represent lower frequencies in the simulation coding accurately, an aperture array was a user chosen variable smaller than the FFT array size.

Another problem that manifests itself is "aliasing", which occurs when high frequency features of a function are

missed by having too few sample points. This means that important information for the function present in the frequencies greater than f_{max} are folded back and appear at lower frequencies. Choosing the distance between sample points smaller than the coherence length of the propagating electromagnetic wave, found theoretically by Equation 2.39, alleviated this problem.

C. SIMULATION DESCRIPTION

1. User defined quantities

For operation of the simulation program, the user had the option of varying several quantities directly from the keyboard:

1. Array size. This can be varied up to a maximum of 512 by 512 points.

2. Aperture sampling points. This can be varied up to a maximum of 512 by 512 points. Ideally it should be smaller than the array size to avoid "tilt" effects, but large enough to avoid "aliasing" effects. A size one-fourth to one-half the array size works best.

3. Aperture shape. A choice of square or circular aperture was provided.

4. Seed. The pseudo-random number generator required a five digit input seed. Table 3.1 lists two quality seeds. All data was obtained using the seed 94377.

5. Turbulence parameter. Any value of C_n^2 above zero can be chosen. Typical values of C_n^2 were 10^{-17} through 10^{-13} $m^{-2/3}$.

6. Wavelength. Any value of wavelength for the electromagnetic wave can be chosen. A value of 0.6 μm was used throughout.

7. Distance. This was the distance from the aperture to the observer. This simulation used 1 km.

8. Aperture width. Adjustable to ensure that "aliasing" effects do not occur for a specific choice of C_n^2. Typically, a 100 mm aperture was chosen with a path length of 1 km.

For the accumulation of multiple data points for statistical purposes, a slightly different version of the program read user choices from a disk file.

2. Aperture Mutual Coherence Function

Computing far-field diffraction patterns of intensity for an electromagnetic wave through the aperture with no turbulence gave the Mutual Coherence Function (MCF) of the aperture. Since we assumed a point source at infinity, the electric field wave front was planar at the aperture. The electric field that passes through had a value of one within the boundaries of the aperture, and a value of zero outside the boundaries. The FFT subroutine computed the diffraction pattern, the electric field present at the image plane. The conjugate square of this electric field was the intensity field that would be seen by an observer. Taking the inverse FFT of the intensity field gave the aperture MCF, which was identical to the autocorrelation of the aperture function.

3. Phase screen generation

Simulation of effects of turbulence requires that the aperture planar electric field be multiplied by the phase screen. Two methods exist to create the phase screens.

a. Classical method

In this method n^2 Gaussian distributed random numbers filled a n by n real array. Since the pseudo-random number generator *RAN* produced uniformly distributed numbers, a subroutine *GAUSS* transformed them to a Gaussian distribution with unit variance, Knuth [Ref. 8]. This subroutine appears in Appendix B. Taking a direct FFT creates two Gaussian distributed arrays, *phaser* and *phasei*, which are the real and imaginary spectral amplitudes of the original random array. Since these two arrays are in spatial frequency space, multiplication by Equation 2.13 occurs. An inverse FFT returns the filter to real space. Because of the symmetry of the filter, discussed later, only one array in real space results, containing all n^2 pieces of information produced by the subroutine *RAN*.

b. Martin and Flatté method

In this second method suggested by Martin and Flatté [Ref. 9], two n by n arrays, called *phaser* and *phasei*, are each filled with n^2 Gaussian distributed random numbers. These are considered the real and imaginary components in spatial frequency space. Filtering occurs as $f^{-11/3}$ then an inverse FFT returns the array to real space. Because no information is lost in filtering, two arrays result, each with n^2 pieces of information. Only one of these arrays need be used for the following simulation code. This method requires twice the random numbers as the previous method but *GAUSS*

produces two random numbers and half as many two-dimensional FFT steps are required. For use in Fresnel propagation, where multiple phase screens exist, this method is certainly the most efficient.

4. Filtering

Phase screens need to be filtered in frequency space to ensure the proper power law form. From Tatarski [Ref. 1] and Martin and Flatté [Ref. 9], the correct form for the filtering function, Φ_f, is

$$\Phi_f = 2\pi\, k^2\, \delta_z\, \Phi_n, \tag{3.1}$$

where k is the wavenumber of the electromagnetic wave, δ_z is the slab thickness, and Φ_n is the power spectral density function. Using Equation 2.13 for the power spectral density function and the definition of wavenumber, $k = 2\pi/\lambda$, gives

$$\Phi_f = 1.303\, (2\pi)^3\, (1/\lambda)^2\, Cn^2\, \delta_z\, f^{-11/3} \tag{3.2}$$

which is the filtering function used in the subroutine *FILTER* in the simulation code. Arrays *phaser* and *phasei* are both multiplied by the square root of Equation 3.2 to obtain a factor for the real and imaginary amplitudes so that the intensity has the proper $f^{-11/3}$ power law.

Although this filtering scheme looks simple, a few subtle points make the process much more complicated than is immediately evident. In fact, perfecting this filtering process constituted the major effort of this thesis.

First of all, most of the relations developed in the background references were in terms of the radian frequency K. The FFT routine used was in terms of the spatial frequency $f = K/2\pi$. Consequently, all relevant equations had to be expressed in this latter form, requiring the tracking down of many 2π terms.

Secondly, due to the two-dimensional isotropy of turbulence, the correct filtering scheme is of a circular nature. This means that the frequency seen in Equation 3.2 is really $f = (f_x^2 + f_y^2)^{1/2}$, which is radial everywhere but at the zero frequency (DC term) where it is has a value of zero [Ref. 9].

For convenience of coding, the zero frequency should be at the center of the arrays *phaser* and *phasei*. But in the FFT algorithm, spatial frequencies are arranged in a manner that is quite different. The zero frequency term is at the upper left corner of the array. The Nyquist "folding" frequency is present as a cross that divides the arrays into four sections. So, prior to filtering, a shuffling of data points is required as discussed by Brigham [Ref. 10] and as seen in Figure 3.1. After filtering, an inverse shuffle returns all frequencies to their previous locations.

Another important consideration in the filtering process was that frequency units be correct according to the physical dimensions of the aperture. This required that the frequency array have a normalization factor of $1/N\Delta$ where N

Figure 3.1 Graphical Presentation Of Two-dimensional FFT Reorganization Required For Conventional Viewing.

is the total number of points across the array, and Δ is the sampling interval (physical units between the aperture array points).

5. Propagation

The Huygens-Fresnel techniques formed the basis for the simulation's treatment of optical propagation. For simplicity this simulation only considered Fraunhofer propagation. The addition of subroutines to handle the dynamics of Fresnel propagation should not be too difficult other than the need for dynamic scaling of the numerical mesh to account for diffraction spreading.

Martin and Flatté provided a procedure to simulate the propagation of a wave in a turbulent medium. [Ref. 9]

1. Add the random turbulence effects to the electric field distribution by meshing the aperture electric field function with a random phase term $\exp(\Psi)$:

$$E(\bar{r}') \rightarrow E(\bar{r}') \exp(\Psi). \tag{3.3}$$

2. Take the Fourier transform of the result of Equation 3.3 to obtain

$$\tilde{E}(\bar{f}') = \mathrm{DFT}\{E(\bar{r}')\}. \tag{3.4}$$

3. Add a term to the result of Equation 3.4 to obtain

$$\tilde{E}(\bar{f}) = \tilde{E}(\bar{f}') \exp\{-\frac{2\pi i R f^2}{k}\} \tag{3.5}$$

which is the exact same relationship as in Equation 2.48 that was described by Roberts as the convolution form of the Fresnel diffraction integral. [Ref. 6]

4. Take the inverse Fourier transform of Equation 3.5 to get an expression for the electric field distribution at the image plane:

$$E(\bar{r}) = IFT\{\tilde{E}(\bar{f})\} \tag{3.6}$$

If one were taking a multiple screen approach to solving for the propagation of the wave as suggested by Martin and Flatté, the entire process would be reiterated for as many segments of length R as required. In this numerical simulation, only one segment was used for all propagations. Adding this iterative section to the code would be a straightforward modification if dynamic array sizes were incorporated. [Ref. 6]

For distances of propagation where the phase front cannot be assumed planar, an additional quadratic term would have to be added at step 3 for the transfer form of the Fresnel diffraction integral, as seen in Equation 2.50.

6. Atmospheric Mutual Coherence Function

We computed the atmospheric mutual coherence function after meshing the random phase screen with the aperture electric field function (after step 1 of the Martin and Flatté method). As with the aperture MCF, the next step was computing the diffraction pattern using the FFT routine, which represents the complex electric field that an observer would see because of the aperture and the turbulence. The effects of turbulence tends to spread out the diffraction pattern so that resolution would be lost for any optical imaging system. Taking the conjugate square of this diffraction pattern gives the intensity field, which negates the additional exponential

term present in Equation 3.5. Taking the inverse FFT of the intensity field gave the total mutual coherence function. Dividing the total MCF by the aperture MCF leaves the atmospheric MCF.

7. Coherence Length

As stated previously, the coherence length, ρ_0, was the distance transverse to the direction of propagation where the spatial coherence of the $\langle E^*E \rangle$ wave dropped to e^{-1}.

For a continuous plot of the atmospheric MCF, the e^{-1} distance could be picked off the curve easily. Due to the discrete nature of the simulation, the MCF curve was not continuous, but a series of points equal to the choice of the pixel width of the aperture array. Therefore this simulation used a linear interpolation routine to pick out values for the coherence length as seen in the results of the following section.

IV. RESULTS

In order to verify that this numerical simulation produced correct results, we had to find ways to check the validity of the code. In the code, presented in Appendix C, there are several checkpoints where the accuracy of the simulation could be verified. The following sections discuss these checkpoints in increasing order of sophistication.

A. APERTURE MCF

An initial point to check the accuracy of the simulation code was to look at the mutual coherence function (MCF) of the aperture. Since the code offered two choices of aperture shape, square and circular, both were verified.

Using the two-dimensional FFT routine, the diffraction patterns for 100 mm wide square and circular apertures were calculated. The square aperture diffraction pattern in Figure 4.1a displays the familiar $[(\sin x)/x]^2$ form discussed by Hecht with the correct maximum value and node spacing. [Ref. 5] The circular aperture diffraction pattern in Figure 4.1b displays the Airy pattern derived from the J_1 Bessel function solution carried out by Hecht. [Ref. 5]

An inverse 2-D FFT of each 100 mm wide aperture shape produced the aperture MCF (identical to aperture autocorrelation). The square aperture autocorrelation in Figure 4.2a is the predicted triangle function. The

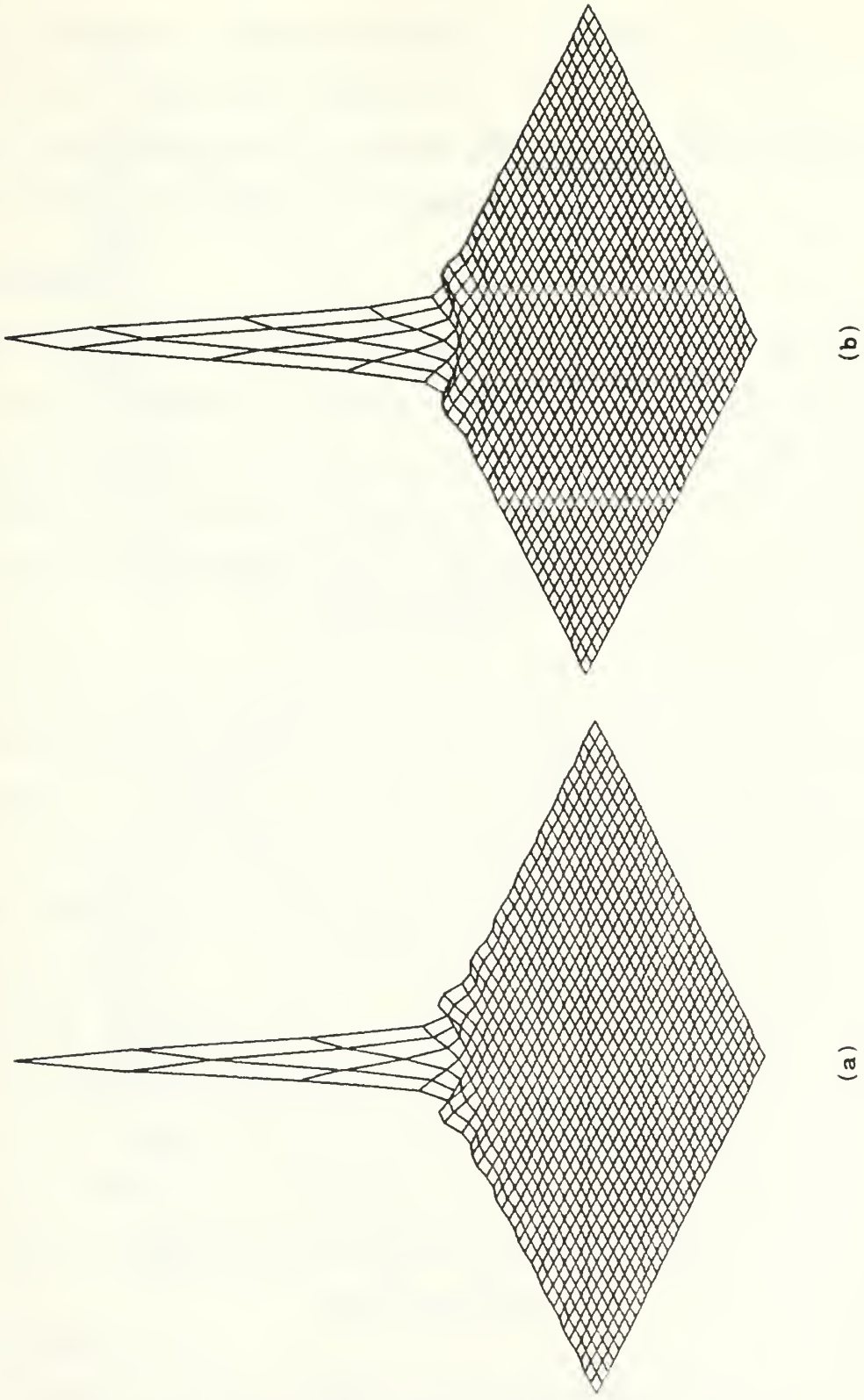


Figure 4.1 Irradiance Diffraction Pattern Resulting From A (a) Square Aperture And A (b) Circular Aperture.

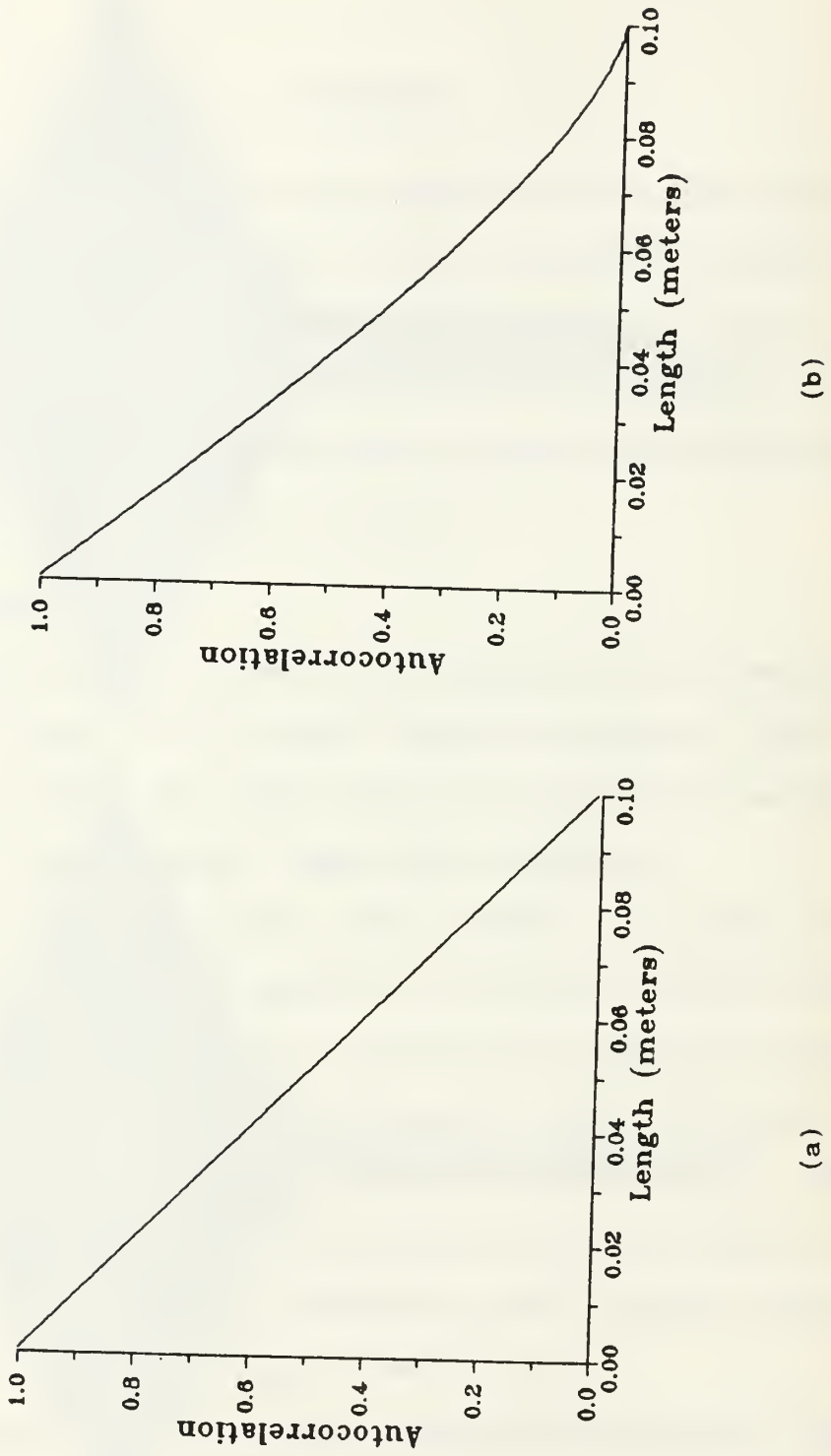


Figure 4.2 Autocorrelation (Aperture Mutual Coherence Function) Of A 100 mm Wide (a) Square Aperture And A 100 mm Wide (b) Circular Aperture

circular aperture autocorrelation in Figure 4.2b is as expected for a "top hat" function.

The algorithm gave the correct MCF results which verified the simulation up through the aperture MCF point.

B. FILTERING

The most crucial portion of the entire simulation code was the correct frequency filtering algorithm. A succinct way of verifying this came from examining the result of taking the logarithm of both sides of Equation 3.3

$$\log\{\Phi_f\} = \log\{1.303(2\pi)^3(1/\lambda)^2 Cn^2 \delta_z f^{-11/3}\}$$

which is

$$\log\{\Phi_f\} = \log\{1.303(2\pi)^3(1/\lambda)^2 Cn^2 \delta_z\} - (11/3)\log\{f\}, \quad (4.1)$$

an equation for a straight line with slope of -11/3.

Figure 4.3 shows a plot of filtering function vs. frequency f with a value 10^{-13} for Cn^2 and a wavelength of 600 nm (red light) using the Martin and Flatté method of filtering discussed earlier. A linear fit to the data points down to the Nyquist frequency had a slope of -3.88. This was a deviation from the predicted value of -11/3 (-3.83) by 1.3%. Therefore, it appears that the filtering is being accomplished correctly. The upturn of points occurs at the Nyquist folding point that is present in the array scheme for the FFT routine.

C. ATMOSPHERIC MCF

An important place to check results of the simulation was at the calculation of the atmospheric MCF. Following the

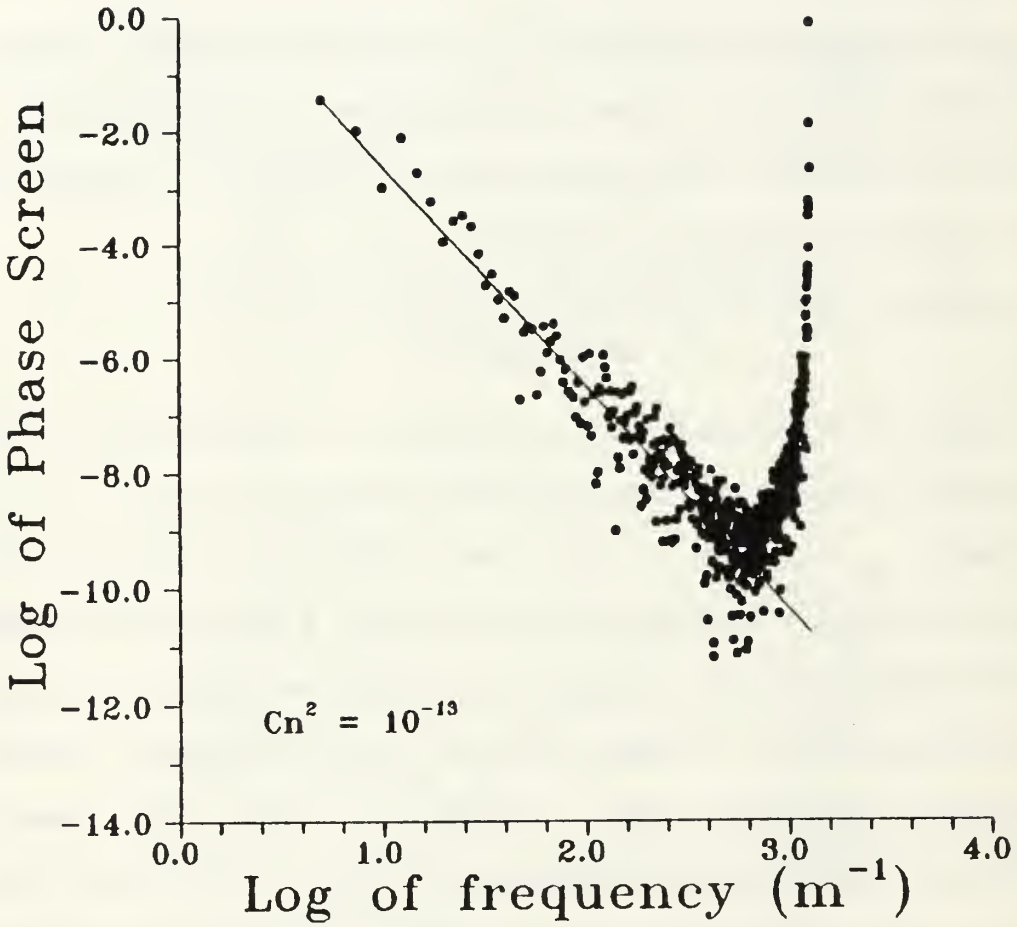


Figure 4.3 Slice Through Phase Screen Function Filtered By The Martin And Flatté Method Showing Its Dependence On Spatial Frequency. Slope Of -11/3 Comes Directly From The Kolmogorov Power Law. The Upturn At Higher Frequencies Is Due To Nyquist Folding Present In The Phase Screen Array Code.

procedure for calculating the atmospheric MCF outlined in section III, first the diffraction pattern was calculated. Figure 4.4a shows the diffraction pattern for a 100 mm wide square aperture that results for red light (λ = 0.6 μm) with $Cn^2 = 10^{-16}$. Notice that the diffraction is only slightly affected from what is shown in Figure 4.1a. As the turbulence structure parameter Cn^2 increased, the diffraction pattern began to spread out more until it no longer resembles the $[(\sin x)/x]^2$ form it had without turbulence, as seen in Figures 4.4b, 4.4c, and 4.4d for values of $Cn^2 = 10^{-15}$, $Cn^2 = 10^{-14}$, and $Cn^2 = 10^{-13}$ respectively. Figure 4.4c shows that modest amounts of turbulence displace the image centroid, due to low frequency tilting of the electric field. Since the diffraction pattern is analogous to the intensity that would be seen at the image plane, this clearly shows the blurring of an image commonly seen looking through a turbulent atmosphere as well as the "dancing" image effect at lower levels of turbulence.

The atmospheric MCF was calculated for different values of turbulence by the method described in section III using the classical method of filtering. These are seen in Figures 4.5 for Cn^2 values of 0, 10^{-16}, 10^{-15}, 10^{-14}, and 10^{-13}. With no turbulence, the atmospheric MCF is a value of 1.0 everywhere, seen as a straight line. With increasing turbulence, the atmospheric MCF curves fall off rapidly to zero.

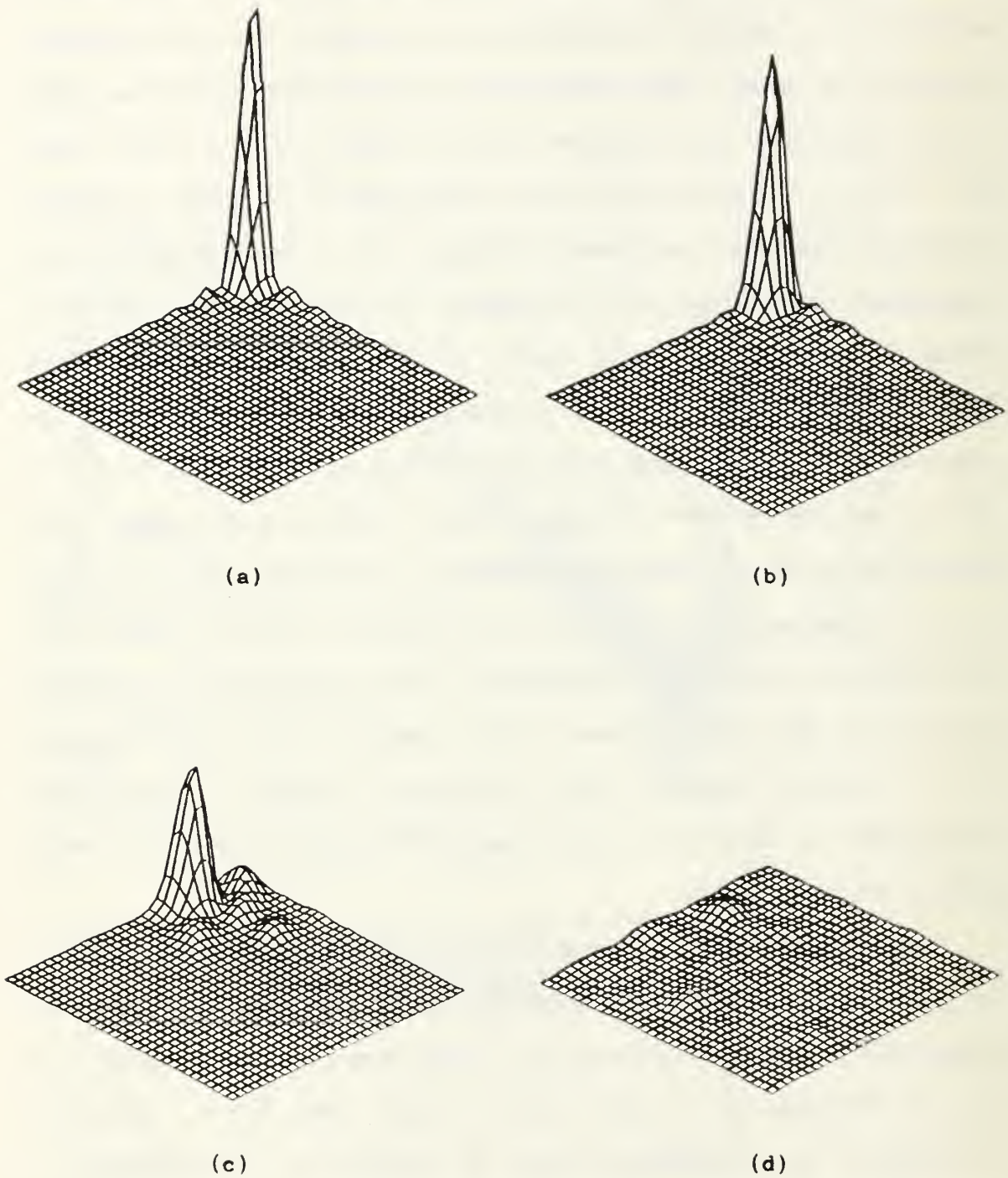


Figure 4.4 Diffraction Patterns For A 100 mm Wide Square Aperture For Four Different Values Of Turbulence: (a) $C_n^2=10^{-16}$ (b) $C_n^2=10^{-15}$ (c) $C_n^2=10^{-14}$ (d) $C_n^2=10^{-13}$. Displacement Of The Beam Centroid Is Obvious In (c) Due To Low Frequency Tilting Of The Electric Field.

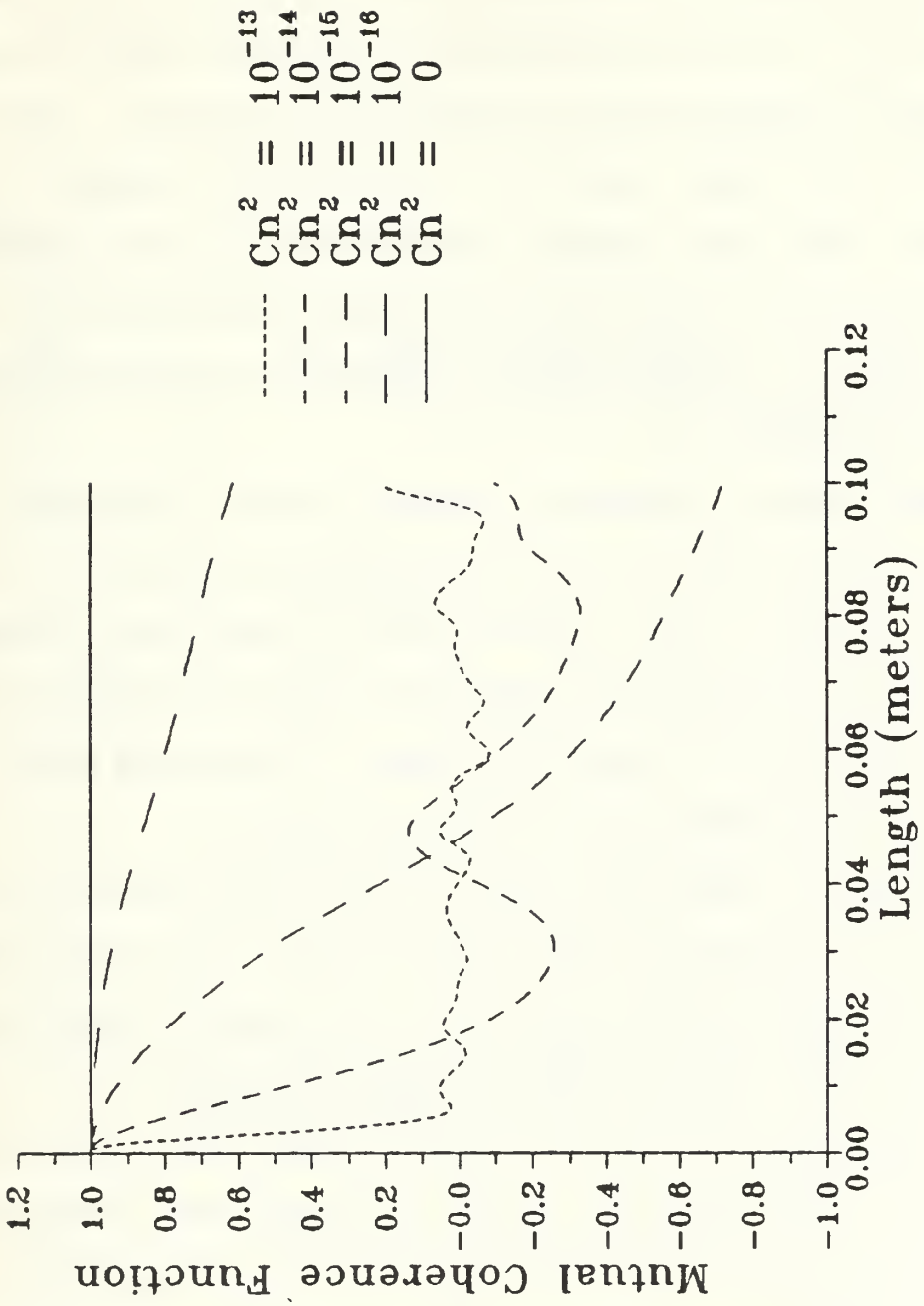


Figure 4.5 Atmospheric Mutual Coherence Function Curves For Five Different Levels Of Turbulence.

D. COHERENCE LENGTH

The e^{-1} point of atmospheric MCF curves give the coherence length, ρ_0. In the simulation, we achieved this by a method of linear interpolation as discussed previously. One should not put too much faith in a statistical sample of one, however. For this reason, a modification of the simulation was made to run several samples of each different set of input parameters so that a better statistical representation of results occurred.

Theoretical coherence lengths were calculated from Equation 2.41 [Ref. 1] for three different values of Cn^2 using R=1000 m and λ=0.6 μm. Several runs of data were produced to investigate the effect of different sizes of both the electric field array and the aperture array. Calculated results are discussed and displayed in following pages.

Figure 4.6 shows the trend for coherence length values as a function of aperture array sample points using the classical method of filtering for $Cn^2 = 10^{-13}$. Ten samples were obtained for each data point. As discussed in section III, aliasing makes the values of ρ_0 too high if there are too few sample points. Using aperture arrays no larger than one-half the width of the main array was done to avoid tilt problems. The curves asymptotically approach coherence length values of 4.18 mm for the 512 x 512 array and 4.49 mm for the 256 x 256 array. These represent errors of 39.3% and 49.7% respectively.

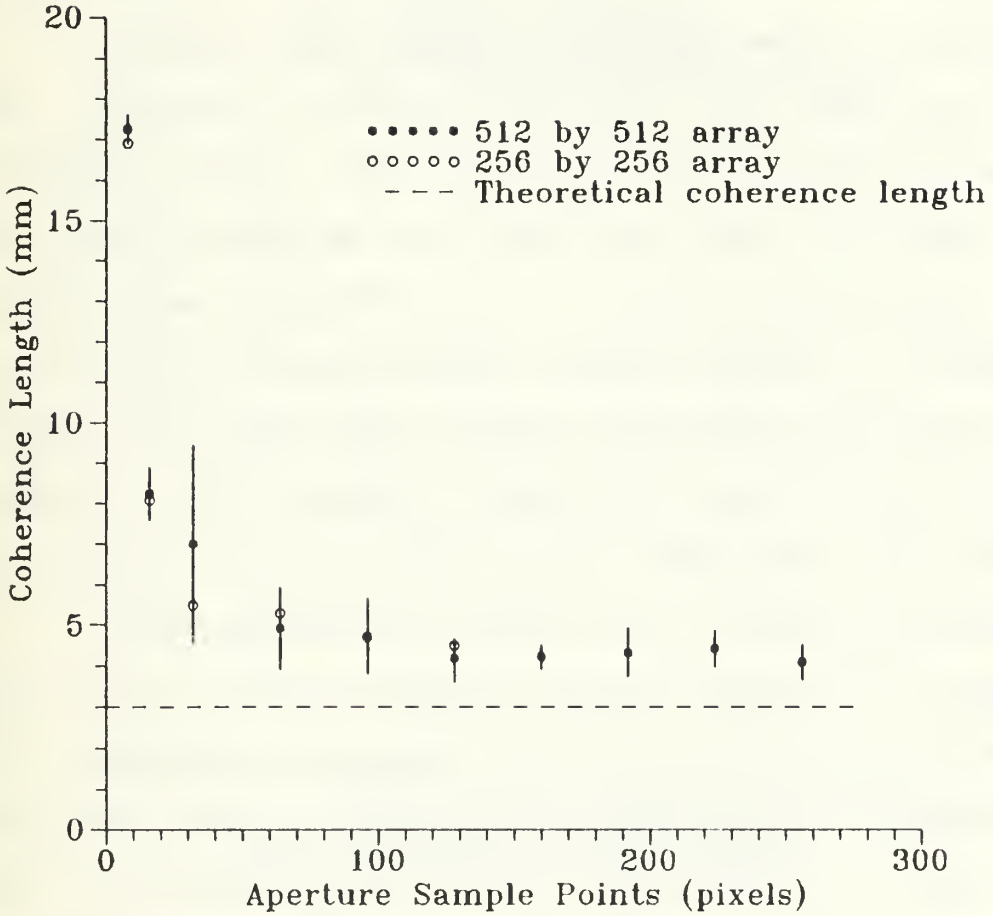


Figure 4.6 Calculated Coherence Length Values For $Cn^2 = 10^{-13}$ Using Different Aperture Array Sizes And Classical Method Of Filtering.
Error Bars Shown Are For Case Of 512 By 512 Array Size, Representing The Standard Deviation For Ten Samples.

Figure 4.7 shows coherence length trends for $Cn^2 = 10^{-14}$ also using the classical method of filtering. Ten samples were obtained for each data point. Results were 15.78 mm for the 512 x 512 array and 18.43 mm for the 256 x 256 array, representing errors of 32.1% and 54.2% respectively.

Figure 4.8 shows coherence length trends for $Cn^2 = 10^{-15}$ also using the classical method of filtering. Ten samples were obtained for each data point. Results were 61.99 mm for the 512 x 512 array and 76.27 mm for the 256 x 256 array, representing errors of 23.3% and 60.3% respectively.

Figure 4.9 shows coherence length trends for $Cn^2 = 10^{-13}$ using the Martin and Flatté method of filtering. Ten samples were obtained for each data point. Results were 4.07 mm for the 512 x 512 array and 4.46 mm for the 256 x 256 array, representing errors of 35.7% and 48.7% respectively.

Figure 4.10 shows coherence length trends for $Cn^2 = 10^{-14}$ also using the Martin and Flatté method of filtering. Ten samples were obtained for each data point. Results were 14.94 mm for the 512 x 512 array and 20.46 mm for the 256 x 256 array, representing errors of 25.0% and 71.2% respectively.

Figure 4.11 shows coherence length trends for $Cn^2 = 10^{-15}$ also using the Martin and Flatté method of filtering. Ten samples were obtained for each data point. Results were 60.74 mm for the 512 x 512 array and 74.04 mm for the 256 x 256 array, representing errors of 27.7% and 55.6% respectively.

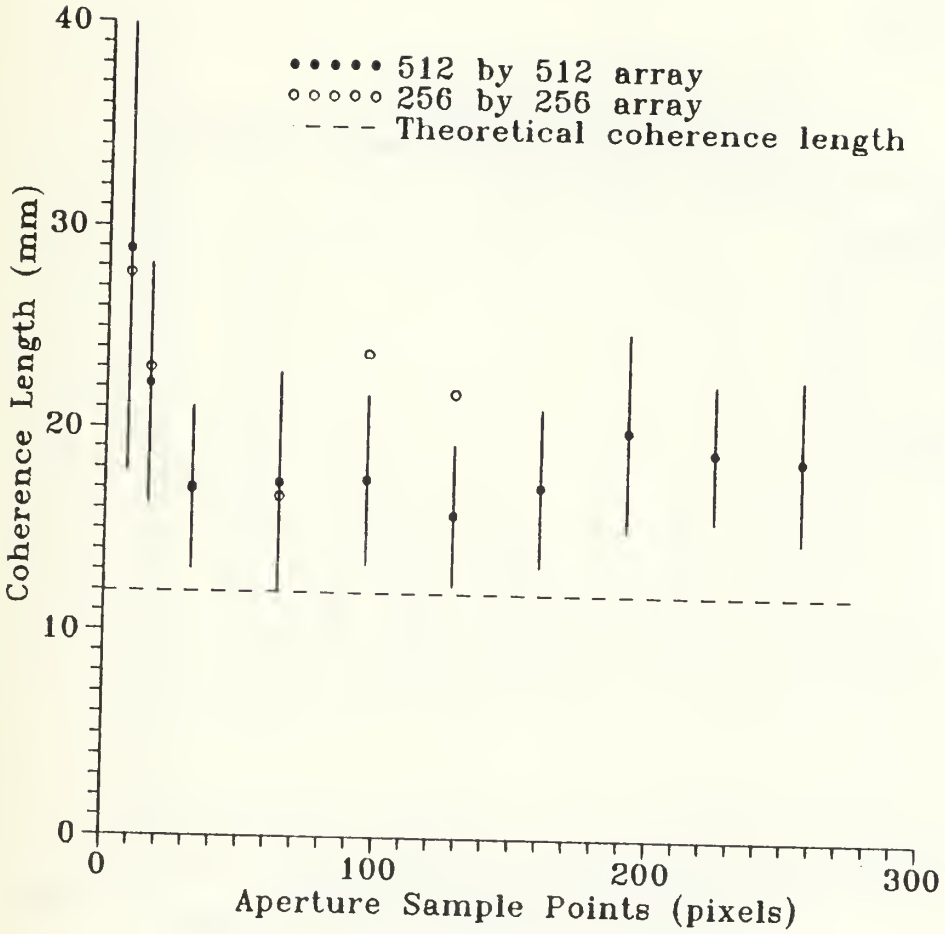


Figure 4.7 Calculated Coherence Length Values For $Cn^2 = 10^{-14}$ Using Different Aperture Array Sizes And Classical Method Of Filtering.
Error Bars Shown Are For Case Of 512 By 512 Array Size, Representing The Standard Deviation For Ten Samples.

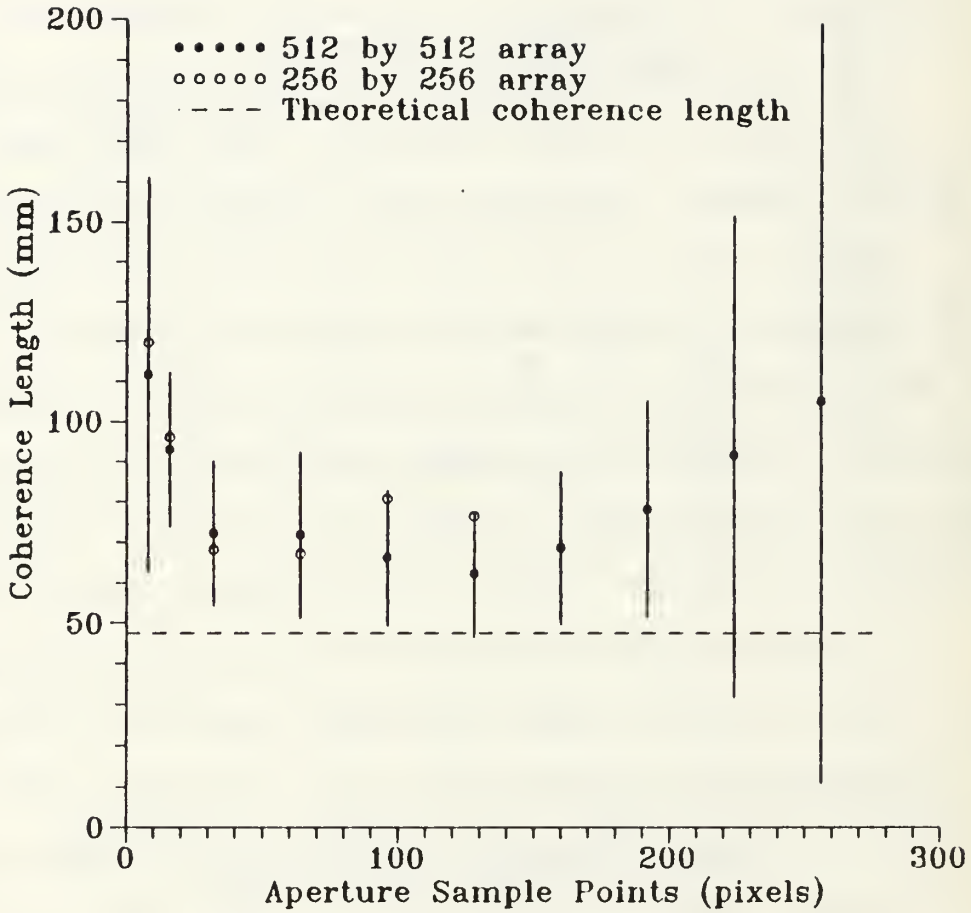


Figure 4.8 Calculated Coherence Length Values For $Cn^2 = 10^{-15}$ Using Different Aperture Array Sizes And Classical Method Of Filtering.
Error Bars Shown Are For Case Of 512 By 512 Array Size, Representing The Standard Deviation For Ten Samples.

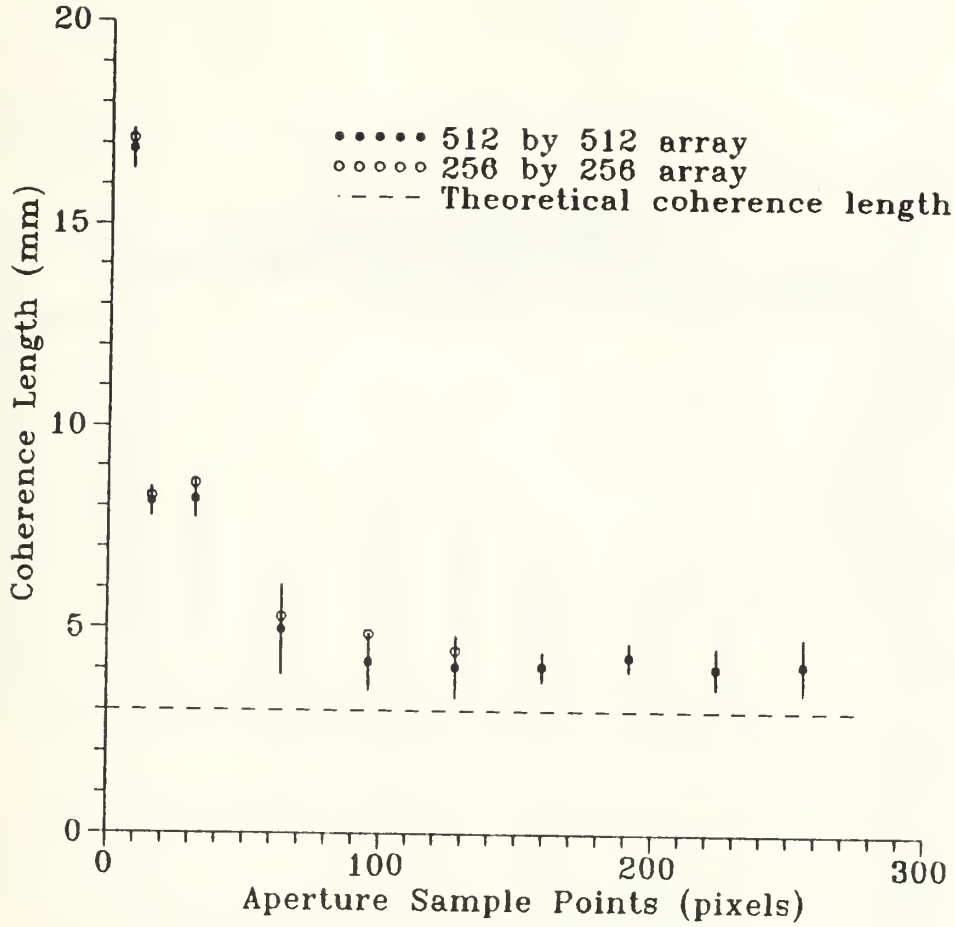


Figure 4.9 Calculated Coherence Length Values For $Cn^2 = 10^{-13}$ Using Different Aperture Array Sizes And Martin & Flatté Method Of Filtering. Error Bars Shown Are For Case Of 512 By 512 Array Size, Representing The Standard Deviation For Ten Samples.

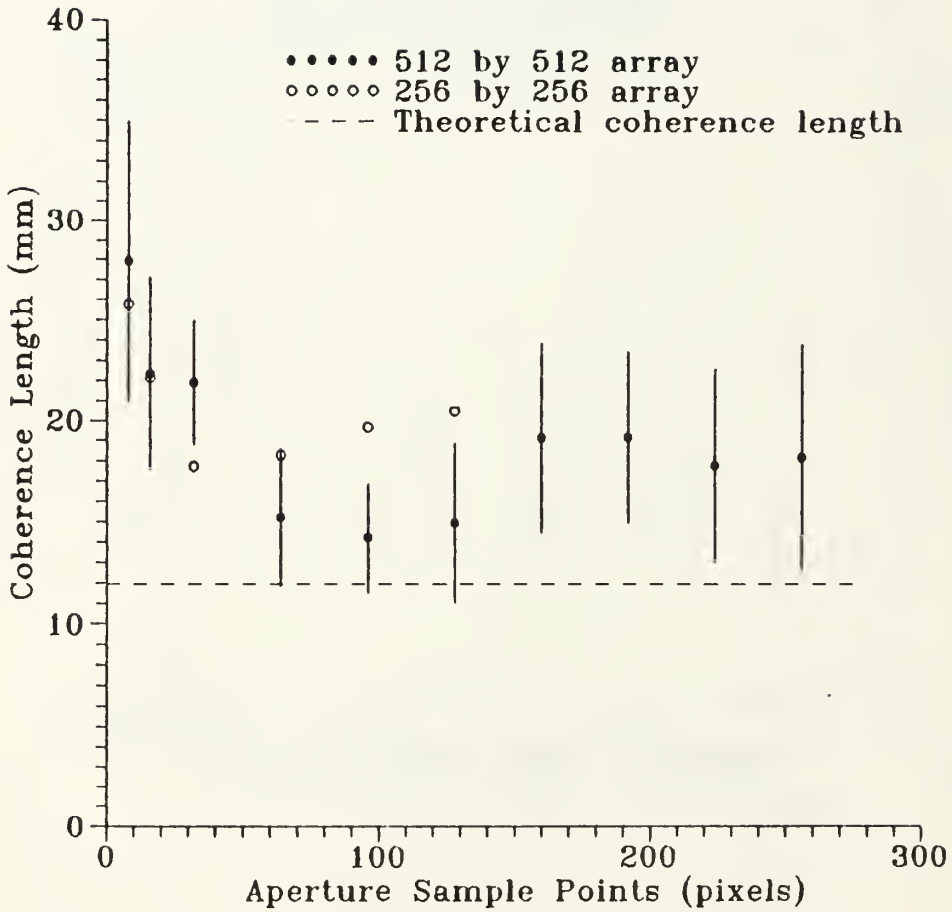


Figure 4.10 Calculated Coherence Length Values For $Cn^2 = 10^{-14}$ Using Different Aperture Array Sizes And Martin & Flatté Method Of Filtering. Error Bars Shown Are For Case Of 512 By 512 Array Size, Representing The Standard Deviation For Ten Samples.

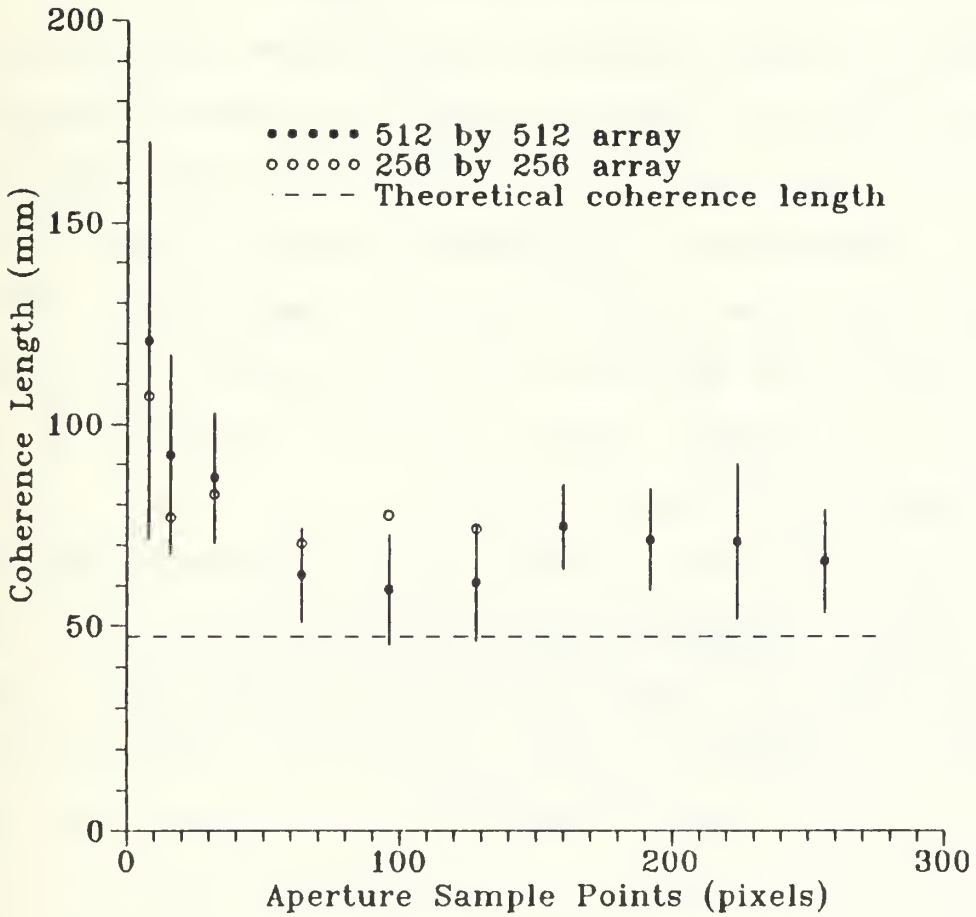


Figure 4.11 Calculated Coherence Length Values For $Cn^2 = 10^{-15}$ Using Different Aperture Array Sizes And Martin & Flatté Method Of Filtering. Error Bars Shown Are For Case Of 512 By 512 Array Size, Representing The Standard Deviation For Ten Samples.

The most accurate results were obtained when the aperture array was one-fourth the size of the electric field array, as seen in Figures 4.6 through 4.11. For this reason all subsequent coherence lengths were calculated using a 512 by 512 array with a 128 by 128 aperture array.

Figure 4.12 shows coherence length values calculated with different values of turbulence using the classical method of filtering. Ten samples were obtained for each calculated value. Although the calculated coherence length values deviated from theoretical values by approximately 30%, a linear fit to the data shows that the slopes were correct, implying a constant error in the algorithm due to underestimating turbulence effects.

Adding a log-normal, random-amplitude screen to the GAUSS subroutine was attempted to improve the simulation results for coherence lengths. Having both phase and amplitude screens present is equivalent to the complete Rytov approximation, Equation 2.31. Eight samples were obtained for each calculated value. Figure 4.13 dramatically shows that this simple addition produced calculated coherence lengths that were within 3% of theoretical values, from a linear fit to the data.

Table 4.1 presents coherence lengths calculated by the simulation using both random phase and amplitude screens in the complete Rytov approximation. All data points were

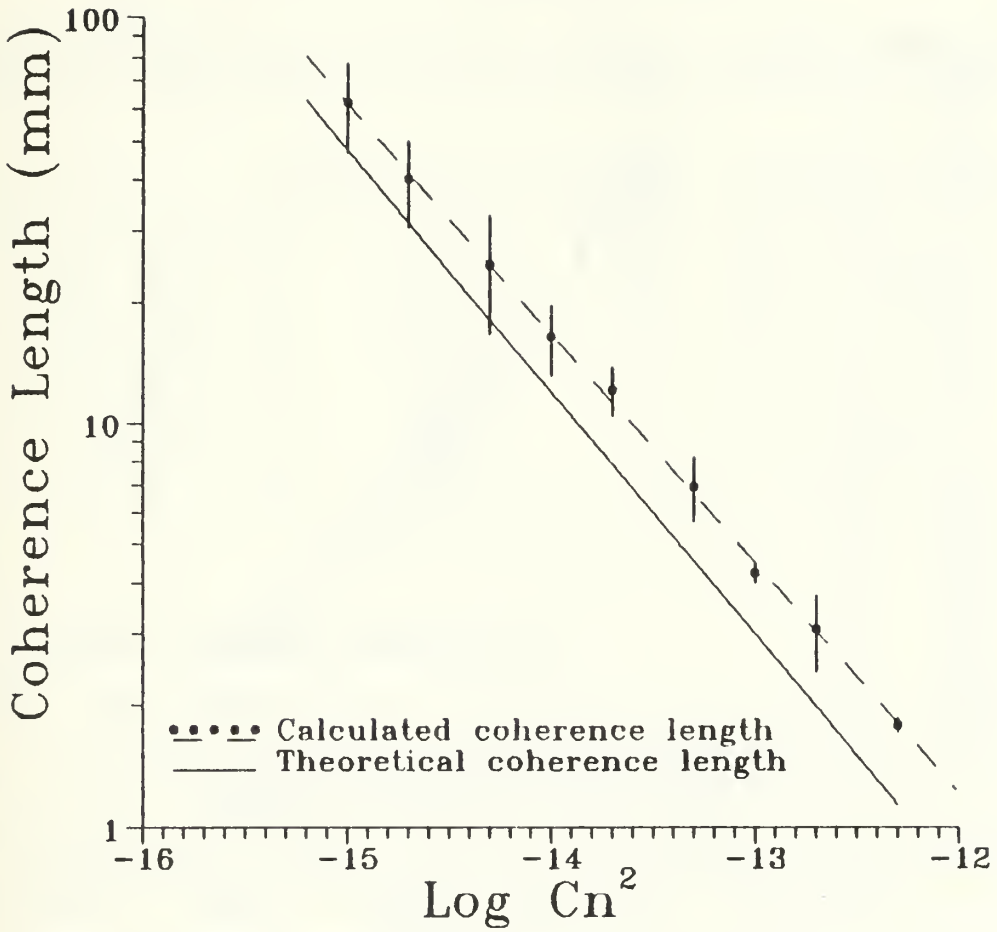


Figure 4.12 Calculated Coherence Length Values As A Function Of Cn^2 With Only Random Phase Screen In The Simulation.
Calculated Values Are About 30% High For All Levels Of Turbulence, From Linear Fit To The Data.
Error Bars Represent Standard Deviation For Ten Samples.

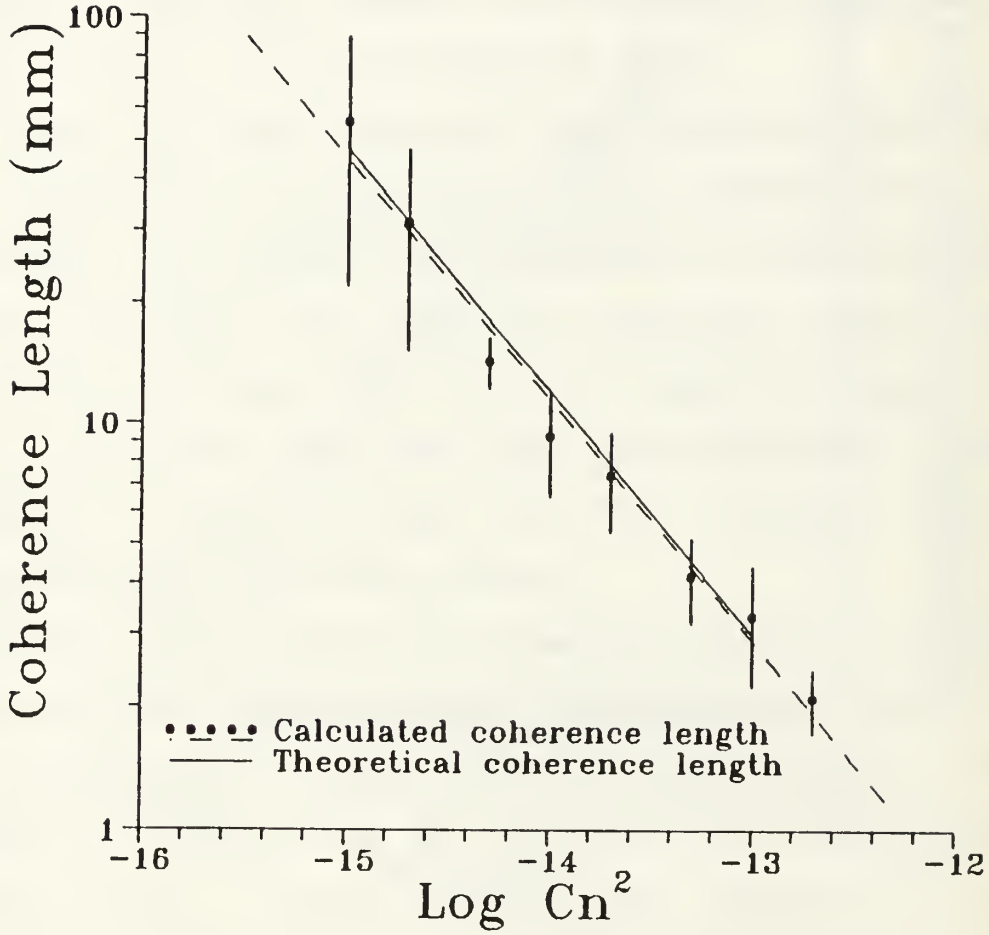


Figure 4.13 Calculated Coherence Length Values As A Function of Cn^2 With Both Random Phase And Amplitude Screens In The Simulation.
Calculated Values Are About 3% High For All Levels Of Turbulence, From Linear Fit To The Data.
Error Bars Represent Standard Deviation For Eight Samples.

obtained using a 512 by 512 array and 128 by 128 aperture array.

TABLE 4.1 COHERENCE LENGTHS

This table shows theoretical values of coherence lengths calculated from Equation 2.39 for eight different values of C_n^2.

C_n^2	theoretical ρ_0 (mm)	calculated ρ_0 (mm)
1×10^{-15}	47.57	50 ± 30
2×10^{-15}	31.39	30 ± 20
5×10^{-15}	18.11	14 ± 2
1×10^{-14}	11.95	9 ± 3
2×10^{-14}	7.88	7 ± 2
5×10^{-14}	4.55	4 ± 1
1×10^{-13}	3.00	3 ± 1
2×10^{-13}	1.98	2.0 ± 0.4

E. INTENSITY VARIANCE SATURATION

The last checkpoint available for verifying proper operation of the simulation was whether the code led to the saturation of intensity variance for increasing turbulence. This should happen because the Rytov assumption of an exponential random term, $\exp(\psi)$, has a magnitude bounded by plus or minus one. Figure 4.14 shows that saturation does occur around $Cn^2 = 10^{-14}$ at a value of unity, as expected. The normalized variance > 1 present at this level of turbulence is due to a "dancing" beam centroid caused by low frequency tilt of the electric field as discussed previously. A decline to unity normalized variance appears in the calculations for higher turbulence values. [Ref. 12]

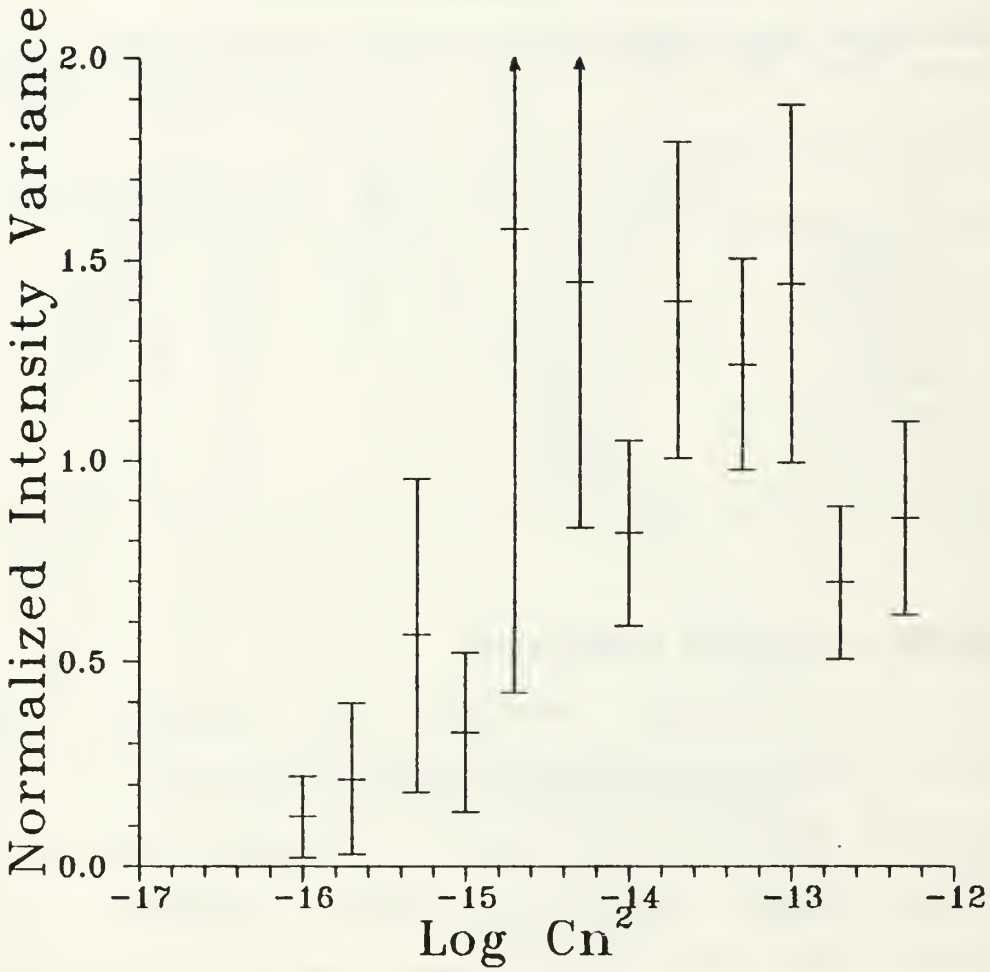


Figure 4.14 Normalized Intensity Variance As A Function Of Cn^2.
Saturation Occurs Around $Cn^2 = 10^{-14}$.
Large Variances Near Saturation Are Due To Beam Centroid Displacement Cause By Low Frequency Tilt.
Error Bars Represent Standard Deviation For Ten Samples.

V. CONCLUSIONS

The program coding discussed in this thesis simulated the propagation of optical waves through a random media using two-dimensional fast Fourier transform (FFT) techniques. The coding used Fraunhofer propagation throughout. An extension to include multi-step Fresnel propagation is straightforward.

Aliasing problems in the FFT were avoided by taking sample points closer together than the theoretical coherence length for a given turbulence structure parameter, C_n^2. Tilt problems in the FFT were suppressed by choosing the aperture array no larger than one-half the size of the FFT array, with one-fourth size giving the best results.

Accuracy of the simulation was verified at various checkpoints within the coding. Aperture diffraction patterns and autocorrelations were compared to expected results. Turbulence effected diffraction patterns and atmospheric mutual coherence functions were presented to show trends for increasing turbulence. Calculated coherence length values were approximately 30% larger than theoretical coherence values when using a 512 by 512 FFT array with a 128 by 128 aperture array. These results were obtained with only a Gaussian distributed random phase screen in the simulation.

The addition of a Gaussian distributed random amplitude screen (for the complete Rytov approximation) to the

simulation brought calculated coherence values to within 3% of theoretical values. This showed that a numerical error was not present in the coding, but that turbulence effects were underestimated without the random amplitude term included. A single step, Fraunhofer algorithm must include the amplitude term to simulate turbulence effects on an optical wave correctly. Unity saturation of the normalized intensity variance occurred for C_n^2 values of approximately 10^{-14}.

Coherence length values used two types of filtering. The classical method and the Martin and Flatté method gave comparable results. The computational efficiency of the Martin and Flatté is significant.

Future endeavors with this simulation should include the incorporation of Fresnel propagation into the coding. Also, further studies with the complete Rytov approximation in the simulation code should be pursued to obtain better statistical results.

APPENDIX A. RANDOM NUMBER GENERATOR SUBROUTINE

```
c-----------------------------------------------------------------------c
      subroutine RAN(iran,r)
c
c   This is the random number generator algorithm from Dr.
c   Harrison's notes, (Ref. 7).
c
c   Variables:
c     iran...input seed value (5 digit integer)
c     r......returned random number uniformly distributed from 0 to 1
c
      iran=iran*99947
      r=0.5 + real(iran)*2.328306e-10
c
      return
      end
c-----------------------------------------------------------------------c
```

APPENDIX B. GAUSSIAN DISTRIBUTION SUBROUTINE

```
c-----------------------------------------------------------------------c
      subroutine GAUSS(narray,iran)
c
c   This subroutine changes uniformly distributed random numbers,
c   provided by subroutine RAN, to Gaussian distributed random
c   numbers with unity variance.  This subroutine is found in
c   Knuth, [Ref. 8].
c
c   Variables:
c     phaser.....real phase screen 2-D array
c     phasei.....imaginary phase screen 2-D array
c     narray.....dimension of the 2-D arrays
c     iran.......input seed value for RAN (5 digit integer)
c     r..........uniformly distributed random numbers
c     x1 & x2....Gaussian distributed random numbers
c
      common /blk2/phaser(512,512),phasei(512,512)
c
      do 10 i=1,narray
        do 10 j=1,narray
   20     call RAN(iran,r)
          v1=2.*r-1.
          call RAN(iran,r)
          v2=2.*r-1.
          s=v1*v1 + v2*v2
          if (s.ge.1.0) goto 20
          scale=sqrt(-2.*alog(s)/s)
          x1=v1*scale
          x2=v2*scale
          phaser(i,j)=x1
          phasei(i,j)=x2
   10 continue
c
      return
      end
c-----------------------------------------------------------------------c
```

APPENDIX C. SIMULATION CODE

```
c-----------------------------------------------------------------------c
c  This code simulates the propagation of a monochromatic optical
c  wave in a turbulent atmosphere.  Two-dimensional FFT routines
c  are used extensively for calculations.  Only Fraunhofer
c  propagation is present in the coding.  Filtering is accomplished
c  by a choice of two methods: (1) classical and (2) the Martin
c  and Flatté method.
c
c  Subroutines:
c     INIT.....Sets electric field arrays to zero.  A value of 0
c              in the call initializes real and imaginary arrays.
c              A value of 1 initializes only the imaginary array.
c     SQUARE...Establishes the planar electric field for the case
c              of a square aperture.
c     CIRCLE...Establishes the planar electric field for the case
c              of a circular aperture.
c     PLOT.....Gives a screen plot of the array chosen in the call
c              using EGA graphics.  Different colors are chosen to
c              represent different orders of magnitude values.  Most
c              calls within this subroutine are NDP Fortran-386
c              specific.
c     XFORM....Takes the 2-D FFT of the two arrays given in the call.
c              Returns values in the same arrays.  Makes calls to
c              subroutine FFT (1-D FFT routine) several times.  A
c              value of -1 in the call is for a direct transform.
c              A value of +1 in the call is for an inverse transform.
c     FFT......1-D FFT routine supplied by Dr. Don Walters.  This
c              routine is used by subroutine XFORM multiple times
c              to accomplish the 2-D transform.
c              This is Dr. Walters FFT
c     MAG......Calculates the electric field magnitude and the
c              intensity field from the real and imaginary electric
c              field arrays.
c     MCFPLOT..Calculates the Mutual Coherence Function (MCF) from
c              the intensity field and then plots it on the screen
c              using EGA graphics.  Many calls within this
c              subroutine are NDP Fortran-386 specific.
c     GAUSS....Creates the phase screens from Gaussian distributed
c              random numbers with unit variance.  Makes calls
c              to subroutine RAN for uniformly distributed random
c              numbers then transforms them to a Gaussian distribution.
c              Elements of this subroutine provided by Knuth. [Ref. 8]
c     RAN......Generates uniformly distributed random numbers.  This
c              algorithm from Harrison [Ref. 7]
```

```
c     FILTER...Filters the phase screens according to the Kolmogorov
c              power law.  A shuffling of array values is necessary
c              for proper filtering.
c     MESH.....Meshes the random phase screen with the electric
c              field.
c
c   Variables:
c     fieldr...Real 2-D electric field array.
c     fieldi...Imaginary 2-D electric field array.
c     fieldro..Real 2-D planar electric field at aperture.
c     phaser...Real 2-D phase screen array.
c     phasei...Imaginary 2-D phase screen array.
c     fieldm...2-D array that represents the electric field magnitude.
c     fieldm2..2-D array that represents the intensity field.
c     narray...Dimension of electric field array
c     nfield...Dimension of aperture array
c     iran.....Input seed for random number generator (5 digit integer).
c     cn2......Refractive index structure parameter.
c     wvl......Wavelength of monochromatic electromagnetic wave.
c     delz.....Physical distance between aperture plane and image plane.
c     width....Physical width of aperture plane.
c     delx.....Physical distance between sample points in aperture
c              array.
c     rhonot...Coherence length calculated by interpolation between
c              points of MCF curve.
c     re.......Real 1-D array used by subroutine FFT.
c     rim......Imaginary 1-D array used by subroutine FFT.
c     fphaser..Real 2-D array used in subroutine FILTER that represents
c              the phase screen in a more convenient form for
c              filtering and viewing.
c     fphasei..Imaginary 2-D array used in subroutine FILTER that
c              represents the phase screen in a more convenient form
c              for filtering and viewing.
c
c
      common /blk1/fieldr(512,512),fieldro(512,512),fieldi(512,512)
      common /blk2/phaser(512,512),phasei(512,512)
      common /blk3/fieldm(512,512),fieldm2(512,512)
c
c   Initialize arrays
c
    1 call INIT(narray,0)
c
c   Input section
c
      write(*,*) 'Enter dimension of array that required.'
      read(*,*) narray
      write(*,*) 'Enter dimension of planar electric field.'
      write(*,*) '(Pixel width of the aperture)'
      read(*,*) nfield
    2 write(*,*) 'Choose aperture shape:  1) SQUARE'
      write(*,*) '                        2) CIRCLE'
```

```
      read(*,*) ichoice
      if (ichoice .lt. 1 .or. ichoice .gt. 2) then
        write(*,*) 'Try again!'
        goto 2
      endif
      write(*,*) 'Enter the seed for the random number generator.'
      write(*,*) '(Value must be a five digit integer)'
      read(*,*) iran
      write(*,*) 'Enter the value of Cn squared.'
      read(*,*) cn2
      write(*,*) 'Enter the wavelength of light.'
      read(*,*) wvl
      write(*,*) 'Enter the distance from aperture to observer.'
      read(*,*) delz
      delz=1000.
      write(*,*) 'Enter the width of the aperture in meters.'
      read(*,*) width
c
      delx=width/real(nfield)
c
c   Load arrays with desired planar electric field
c
      if (ichoice .eq. 1) call SQUARE(narray,nfield)
      if (ichoice .eq. 2) call CIRCLE(narray,nfield)
c
c   Plot planar electric field
c
      call PLOT(fieldr,narray,1)
c
c   Take Fourier transform of the field
c
      call XFORM(fieldr,fieldi,narray,delx,-1.)
c
c   Calculate magnitude of transformed field then plot it
c
      call MAG(narray)
c
      call PLOT(fieldm,narray,1)
c
c   Set imaginary portion of field to zero
c
      call INIT(narray,1)
c
c   Take Fourier transform of field intensity
c
      call XFORM(fieldm2,fieldi,narray,delx,+1.)
c
c   Calculate and plot MCF of aperture
c
      call MCFPLOT(narray,delx,nfield,0)
c
c   Load phase screen arrays with Gaussian random numbers
```

```
c
      call GAUSS(narray,iran)
c
c   Transform phase screen to frequency space
c
      call XFORM(phaser,phasei,narray,sqrt(real(narray)),-1.)
c     call PLOT(phaser,narray,1)
c     call PLOT(phasei,narray,1)
c
c   Filter phase screen using Kolmogorov spectrum idea
c
      call FILTER(narray,cn2,wvl,delx,delz)
c     call PLOT(phaser,narray,1)
c     call PLOT(phasei,narray,1)
c
c   Transform phase screen back to real space
c
      call XFORM(phaser,phasei,narray,sqrt(real(narray)*delx),+1.)
c     call PLOT(phaser,narray,1)
c     call PLOT(phasei,narray,1)
c
c   Mesh phase screen with planar electric field
c
      call INIT(narray,1)
      call MESH(narray)
c
      call PLOT(fieldr,narray,1)
c
c   Take fourier transform of this electric field
c
      call XFORM(fieldr,fieldi,narray,delx,-1.)
c
      call MAG(narray)
c
      call PLOT(fieldm,narray,1)
c
c   Reset imaginary portion of field to zero
c
      call INIT(narray,1)
c
c   Take Fourier transform to get diffraction pattern w/ turbulence.
c
      call XFORM(fieldm2,fieldi,narray,delx,+1.)
c
c   Calculate atmospheric MCF and plot it.
c
      call MCFPLOT(narray,delx,nfield,1)
c
      goto 1
      stop
      end
```

```
c-----------------------------------------------------------------c
      subroutine INIT(narray,k)
c
      common /blk1/fieldr(512,512),fieldro(512,512),fieldi(512,512)
c     do 10 i=1,narray
        do 10 j=1,narray
          if (k .eq. 0) then
            fieldr(i,j)=0.0
            fieldro(i,j)=0.0
          endif
          fieldi(i,j)=0.0
   10 continue
c
      return
      end
c-----------------------------------------------------------------c
      subroutine MAG(narray)
c
      common /blk1/fieldr(512,512),fieldro(512,512),fieldi(512,512)
      common /blk3/fieldm(512,512),fieldm2(512,512)
c
      do 10 i=1,narray
        do 10 j=1,narray
          fieldm2(i,j)=fieldr(i,j)**2 + fieldi(i,j)**2
          fieldm(i,j)=sqrt(fieldm2(i,j))
   10 continue
c
      return
      end
c-----------------------------------------------------------------c
      subroutine SQUARE(narray,nfield)
c
      common /blk1/fieldr(512,512),fieldro(512,512),fieldi(512,512)
c
      do 10 i=narray/2+1-nfield/2,narray/2+nfield/2
        do 10 j=narray/2+1-nfield/2,narray/2+nfield/2
          fieldr(i,j)=1.0
          fieldro(i,j)=1.0
   10 continue
c
      return
      end
c-----------------------------------------------------------------c
      subroutine CIRCLE(narray,nfield)
c
      common /blk1/fieldr(512,512),fieldro(512,512),fieldi(512,512)
      narray2=narray/2
      nfield2=nfield/2
c
      do 10 i=narray2+1-nfield2,narray2+nfield2
        do 20 j=narray2+1-nfield2,narray2+nfield2
```

```
          radius2=(real(i)-real(narray2))**2 +
     *            (real(j)-real(narray2))**2
          if (radius2 .lt. real(nfield2)**2) then
            fieldr(i,j)=1.0
            fieldro(i,j)=1.0
          endif
   20   continue
   10 continue
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine PLOT(field,narray,k)
c
      dimension field(512,512)
      dimension ndex(20)
      data ndex/4,4,12,12,14,14,10,10,2,2,3,3,9,9,1,1,8,8,0,0/
c
      fmax=0.0
      do 100 i=1,narray
        do 100 j=1,narray
          x=field(i,j)
          if (x.gt.fmax) then
            fmax=x
          endif
  100 continue
c
      call set_video_mode(16)
      call ega_set_mode_4
c
      do 110 i=1,narray/k
        do 110 j=1,narray/k
          ix=i
          iy=j
          xmax=alog10(field(i,j)/fmax)
          index=8*abs(xmax)+1
          if (index.ge.21) then
            icolor=0
          else
            icolor=ndex(index)
          endif
          call ega_put_pixel(ix,iy,icolor)
  110 continue
c
      call pause
      call set_video_mode(3)
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine MCFPLOT(narray,delx,nfield,iaprture)
```

```
c
      common /blk3/fieldm(512,512),fieldm2(512,512)
      dimension apmcf(513),ymcf(513)
      logical flag
      efold=0.36787944
      rhonot=0.
      flag=.true.
c
      do 10 i=1,513
        if (iaprture .eq. 0) then
          apmcf(i)=0.0
        endif
        ymcf(i)=0.0
   10 continue
c
      fmcf=0.0
      do 20 i=1,narray
        do 20 j=1,narray
          xmcf=fieldm2(i,j)
          if (xmcf.gt.fmcf) then
            fmcf=xmcf
          endif
   20 continue
c
      do 30 i=1,nfield
        if (iaprture .eq. 0) then
          apmcf(i)=fieldm2(1,i)/fmcf
          ymcf(i)=apmcf(i)
        else
          ymcf(i)=fieldm2(1,i)/(apmcf(i)*fmcf)
        endif
   30 continue
c
      do 40 i=1,nfield
        if (iaprture.eq.1) then
          if (flag) then
            if (ymcf(i).lt.efold) then
              rhonot=real(i-1)*delx
     *               + (efold-ymcf(i))*delx/(ymcf(i-1)-ymcf(i))
              flag=.false.
            endif
          endif
        endif
   40 continue
c
      call set_video_mode(16)
      call ega_set_mode_4
      call locate(20,1)
      if (iaprture .eq. 0) then
        call write_string('Aperture MCF vs. Length')
      else
        call write_string('Atmospheric MCF vs. Length')
```

```
      endif
      call locate(1,1)
      call write_string('1.0')
      call locate(25,24)
      call write_string('length')
c     call locate(37,24)
c     call write_string(nfield)
      call ega_draw_line(40,20,40,320,15)
      call ega_draw_line(40,320,600,320,15)
c
      ix=40
      iy=20
      do 50 i=1,nfield
        ix2=40+560*i/nfield
        iy2=20+int(300.*(1.-ymcf(i+1)))
        call ega_draw_line(ix,iy,ix2,iy2,15)
        ix=ix2
        iy=iy2
   50 continue
c
      call pause
      call set_video_mode(3)
c
      write(*,*) 'The coherence length is ',rhonot,' meters.'
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine XFORM(fieldr,fieldi,narray,fftnorm,sign)
c
      common /blk4/re(512),rim(512)
      dimension fieldr(512,512),fieldi(512,512)
      data re/512*0./,rim/512*0./
c
      m=int(alog(real(narray))/alog(2.))
c
      do 30 i=1,narray
        do 40 j=1,narray
          re(j)=fieldr(i,j)
          rim(j)=fieldi(i,j)
   40   continue
        call FFT(m,fftnorm,sign)
        do 50 j=1,narray
          fieldr(i,j)=re(j)
          fieldi(i,j)=rim(j)
   50   continue
   30 continue
      do 60 j=1,narray
        do 70 i=1,narray
          re(i)=fieldr(i,j)
          rim(i)=fieldi(i,j)
```

```
   70   continue
        call FFT(m,fftnorm,sign)
        do 80 i=1,narray
          fieldr(i,j)=re(i)
          fieldi(i,j)=rim(i)
   80   continue
   60 continue
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine FFT(m,fftnorm,sign)
c
      common /blk4/re(512),rim(512)
      pi=3.141592653589792*sign
      n=2**m
      n1=n-1
      j=1
      do 200 i=1,n1
        if (i.lt.j) then
          t=re(j)
          re(j)=re(i)
          re(i)=t
          t=rim(j)
          rim(j)=rim(i)
          rim(i)=t
        endif
        k=n/2
        do 201 while (k.lt.j)
          j=j-k
          k=k/2
  201   continue
        j=j+k
  200 continue
      le=1
      do 202 l=1,m
        le1=le
        le=le+le
        ure=1.
        uim=0.
        ang=pi/le1
        wre=cos(ang)
        wim=sin(ang)
        do 203 j=1,le1
          do 204 i=j,n,le
            ip=i+le1
            tre=re(ip)*ure-rim(ip)*uim
            tim=re(ip)*uim+rim(ip)*ure
            re(ip)=re(i)-tre
            rim(ip)=rim(i)-tim
            re(i)=re(i)+tre
            rim(i)=rim(i)+tim
```

```
  204     continue
          t=ure*wre-uim*wim
          uim=ure*wim+uim*wre
          ure=t
  203   continue
  202 continue
      if (sign.gt.0.0) then
        pts=1.0/real(n*fftnorm)
      else
        pts=fftnorm
      endif
      do 205 i=1,n
        re(i)=re(i)*pts
        rim(i)=rim(i)*pts
  205 continue
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine GAUSS(narray,iran)
c
      common /blk2/phaser(512,512),phasei(512,512)
c
      do 10 i=1,narray
        do 10 j=1,narray
   20     call RAN(iran,r)
          v1=2.*r-1.
          call RAN(iran,r)
          v2=2.*r-1.
          s=v1*v1 + v2*v2
          if (s.ge.1.0) goto 20
          scale=sqrt(-2.*alog(s)/s)
          x1=v1*scale
          x2=v2*scale
          phaser(i,j)=x1
          phasei(i,j)=x2
   10 continue
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine RAN(iran,r)
c
      kran=99947
c
      iran=iran*kran
      r=0.5 + real(iran)*2.328306e-10
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine FILTER(narray,cn2,wvl,delx,delz)
```

```
c
      common /blk2/phaser(512,512),phasei(512,512)
      dimension fphaser(512,512),fphasei(512,512)
      dimension aaa(512,512)
      pi=3.141592653589792
      tpi=2.*pi
      narray2=narray/2
      npivot=narray2+1
      power=-11./6.
      factor=sqrt(1.303*(tpi**3)*cn2*delz)/wvl
      factor2=(tpi/real(delx*narray))**power
c
      do 10 i=1,narray
        do 10 j=1,narray
          fphaser(i,j)=0.
          fphasei(i,j)=0.
   10 continue
c
      do 20 i=1,narray
        do 20 j=1,narray
          if (i.le.npivot) then
            if (j.le.npivot) then
              fphaser(i-1+narray2,j-1+narray2)=phaser(i,j)
              fphasei(i-1+narray2,j-1+narray2)=phasei(i,j)
            else
              fphaser(i-1+narray2,j-1-narray2)=phaser(i,j)
              fphasei(i-1+narray2,j-1-narray2)=phasei(i,j)
            endif
          else
            if (j.le.npivot) then
              fphaser(i-1-narray2,j-1+narray2)=phaser(i,j)
              fphasei(i-1-narray2,j-1+narray2)=phasei(i,j)
            else
              fphaser(i-1-narray2,j-1-narray2)=phaser(i,j)
              fphasei(i-1-narray2,j-1-narray2)=phasei(i,j)
            endif
          endif
   20 continue
c
      do 30 i=1,narray
        do 30 j=1,narray
          freq=sqrt(real(i-narray2)**2 + real(j-narray2)**2)
          if (i.eq.narray2.and.j.eq.narray2) then
            fphaser(i,j)=0.
            fphasei(i,j)=0.
          else
            fphaser(i,j)=fphaser(i,j)*factor*factor2*(freq)**power
            fphasei(i,j)=fphasei(i,j)*factor*factor2*(freq)**power
          endif
   30 continue
```

```
c
      do 40 i=1,narray
        do 40 j=1,narray
          if (i.lt.narray2) then
            if (j.lt.narray2) then
              phaser(i+1+narray2,j+1+narray2)=fphaser(i,j)
              phasei(i+1+narray2,j+1+narray2)=fphasei(i,j)
            else
              phaser(i+1+narray2,j+1-narray2)=fphaser(i,j)
              phasei(i+1+narray2,j+1-narray2)=fphasei(i,j)
            endif
          else
            if (j.lt.narray2) then
              phaser(i+1-narray2,j+1+narray2)=fphaser(i,j)
              phasei(i+1-narray2,j+1+narray2)=fphasei(i,j)
            else
              phaser(i+1-narray2,j+1-narray2)=fphaser(i,j)
              phasei(i+1-narray2,j+1-narray2)=fphasei(i,j)
            endif
          endif
   40 continue
c
      return
      end
c-----------------------------------------------------------------------c
      subroutine MESH(narray)
c
      common /blk1/fieldr(512,512),fieldro(512,512),fieldi(512,512)
      common /blk2/phaser(512,512),phasei(512,512)
c
      do 10 i=1,narray
        do 10 j=1,narray
          ercosphi=fieldro(i,j)*cos(phaser(i,j))*exp(phasei(i,j))
          eicosphi=fieldi(i,j)*cos(phaser(i,j))*exp(phasei(i,j))
          ersinphi=fieldro(i,j)*sin(phaser(i,j))*exp(phasei(i,j))
          eisinphi=fieldi(i,j)*sin(phaser(i,j))*exp(phasei(i,j))
          fieldr(i,j)=ercosphi-eisinphi
          fieldi(i,j)=ersinphi+eicosphi
   10 continue
c
      return
      end
```

LIST OF REFERENCES

1. Walters, D. L., "Propagation through Atmospheric Turbulence", *High Energy Laser Propagation Handbook*, Chapter 5, Optimetrics, Inc., Ann Arbor, Michigan, 1983.

2. Tatarski, V. I., *Wave Propagation in a Turbulent Medium*, McGraw-Hill, New York, 1961.

3. Clifford, S. F., "The Classical Theory of Wave Propagation in a Turbulent Medium", *Topics in Applied Physics, Laser Beam Propagation in the Atmosphere*, v. 25, Chapter 2, Springer-Verlag, 1978.

4. Lutomirski, R. F. and Yura, H. T., "Propagation of a Finite Optical Beam in an Inhomogeneous Medium", *Applied Optics*, v. 10, No. 7, p. 1656, July 1971.

5. Hecht, E., *Optics*, Second Edition, Chapter 11, pp. 472-515, Addison-Wesley, Reading, Massachusetts, 1987.

6. Roberts, P. H., "A Wave Optics Propagation Code", The Optical Sciences Company Report TR-760, pp. 1-15, December 1986.

7. Harrison, D. E., PH 3911 *Simulation of Physical Systems*, class notes, Naval Postgraduate School, 1986.

8. Knuth, D. E., *Seminumerical Algorithms: The Art of Computer Programming*, v. 2, Chapter 3, Addison-Wesley, Reading, Massachusetts, 1981.

9. Martin, J. M., and Flatté, S. M., "Intensity Images and Statistics for Numerical Simulation of Wave Propagation in 3-D Random Media", *Applied Optics*, v. 27, No. 11, pp. 2111-2126, 1988.

10. Brigham, E. O., *The Fast Fourier Transform*, pp. 232-271, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1988.

11. Bevington, P. R., *Data Reduction and Error Analysis for the Physical Sciences*, pp. 314-315, McGraw-Hill Book Company, New York, 1969.

12. Jakeman, E. and Pusey, P. N., "Significance of K Distributions in Scattering Experiments", *Physical Review Letters*, v. 40, no. 9, pp. 546-550, Feb. 1978.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Professor Donald L. Walters, Code 61We Department of Physics Naval Postgraduate School Monterey, California 93943-5000	10
4.	Professor Edmund A. Milne, Code 61Mn Department of Physics Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor K. E. Woehler, Code 61 Chairman, Department of Physics Naval Postgraduate School Monterey, California 93943-5000	1
6.	Dr. Barry Hogge Weapons Laboratory/AR Kirtland Air Force Base, New Mexico 87117	1
7.	Dr. Robert Fugate Weapons Laboratory/ARCA Kirtland Air Force Base, New Mexico 87117	3
8.	Dr. Bobby Junkers Office of Naval Research Code 111 Arlington, Virginia 2217-5000	1
9.	Dr. Kenneth Johnston Naval Research Laboratory Code 4130 Washington, D.C. 20375	1
10.	Captain Nelson Torres, USA Joint Electronic Warfare Center Computer Applications Directorate San Antonio, Texas 78243-5000	4

11. **Lieutenant Jeffrey L. Turner, USN** **2**
Joint Electronic Warfare Center
Computer Applications Directorate
San Antonio, Texas 78243-5000

12. Research Administration 1
Code 012
Naval Postgraduate School
Monterey, CA 93943

Thesis
T9546 Turner
c.1 A simulation of optical propagation through atmospheric turbulence using two-dimensional Fourier transform techniques.

A simulation of optical propagation thro

3 2768 000 90769 5
DUDLEY KNOX LIBRARY